

Considered Scenario

Must have scenario

1. S2(5/6)1: - Has single technology, **Dedicated Access Network for each domain (RAN) and Shared Core**. **Track Parallel to Road**
2. S4(5/6)1: - Has single technology, **Shared Access Network and Shared Core**. **Track Parallel to Road**
3. S4(5/6)4: - Has single technology, **Shared Access Network and Shared Core**. **Track Perpendicular to Road**
4. S1(5/6)1: - Has single technology, **Dedicated Access Network for each domain (RAN) and Dedicated Core**. **Track Parallel to Road**
5. S1(5/6)4: - Has single technology, **Dedicated Access Network for each domain (RAN) and Dedicated Core**. **Track Perpendicular to Road**

Nice to have scenario

1. S4(5/6)3: - Has single technology, **Shared Access Network and Shared Core**. **Track Parallel to Road (Tunnel, have more than 30 meters of Distance between Railways and Road)**
2. S6(5/6)1: - Has different (different frequency) technology, **Dedicated Access Network for each domain (RAN) and Shared Core**. **Track Parallel to Road**
3. S441: - Has single technology, **Shared Access Network and Shared Core**. **Track Parallel to Road (Highway and High-speed Train)**
4. S241: - Has single technology, **Dedicated Access Network for each domain (RAN) and Shared Core**. **Track Parallel to Road**
5. S641: - Has different (different frequency) technology, **Dedicated Access Network for each domain (RAN) and Shared Core**. **Track Parallel to Road (Highway and High-speed Train)**
6. S8(5/6)1: - Has different (different frequency) technology, **Shared Access Network and Shared Core**. **Track Parallel to Road (Highway and High-speed Train)**

CODE

1. S2(5/6)1: - Has single technology, **Dedicated Access Network for each domain (RAN) and Shared Core.** **Track Parallel to Road**

```
#!/usr/bin/python
```

```
'UPDATED Example for Handover for PARALLEL Difference Access and Shared Core'
```

```
import sys
import time
```

```
from mininet.log import setLogLevel, info
from mn_wifi.cli import CLI
from mn_wifi.net import Mininet_wifi
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.log import setLogLevel, info
from mininet.link import TCLink, Intf
from subprocess import call
```

```
def topology(args):
```

```
    "Create a network."
    #net = Mininet_wifi()
```

```
    net = Mininet_wifi( topo=None,
                        build=False,
                        ipBase='192.168.0.0/24')
```

```
    info( '*** Adding controller\n' )
    c1=net.addController(name='c1',
                        controller=RemoteController,
                        ip='127.0.0.1',
                        protocol='tcp',
                        port=6653)
```

```
    kwargs = {'protocols':'OpenFlow13','txpower':'26dBm','range': 50 }# 'failMode': 'standalone', 'datapath': 'user'}
```

```
    info( '*** Add switches\n' )
    s11 = net.addSwitch('s11', cls=OVSKernelSwitch,protocols="OpenFlow13")
    s22 = net.addSwitch('s22', cls=OVSKernelSwitch,protocols="OpenFlow13")
    s33 = net.addSwitch('s33', cls=OVSKernelSwitch,protocols="OpenFlow13")
```

```
    info( "*** Creating nodes\n" )
    Tra1_args, Tra3_args, Car1_args = dict(), dict(), dict()
    if '-s' in args:
        Tra1_args['position'], Tra3_args['position'], Car1_args['position'] = '10,110,0', '150.110.0', '10,10,0'
```

```
    Tra1 = net.addStation('Tra1', mac='00:00:00:00:00:01', ip='192.168.7.101/24', position='10,110,0', **Tra1_args)
    Tra2 = net.addStation('Tra2', mac='00:00:00:00:00:03', ip='192.168.7.102/24', position='10,110,0')
    Tra3 = net.addStation('Tra3', mac='00:00:00:00:00:05', ip='192.168.7.103/24', position='150,110,0', **Tra3_args)
    Tra4 = net.addStation('Tra4', mac='00:00:00:00:00:09', ip='192.168.7.105/24', position='150,110,0')
```

```
    Car1 = net.addStation('Car1', mac='00:00:00:00:00:02', ip='192.168.0.201/24', position='10,10,0', **Car1_args)
    Car2 = net.addStation('Car2', mac='00:00:00:00:00:04', ip='192.168.0.202/24', position='10,10,0')
    Car3 = net.addStation('Car3', mac='00:00:00:00:00:06', ip='192.168.0.203/24', position='150,10,0')
    Car4 = net.addStation('Car4', mac='00:00:00:00:00:10', ip='192.168.0.205/24', position='150,10,0')
```

```
    info( '*** Add hosts\n' )
    CarServer = net.addHost('CarServer', cls=Host, ip='192.168.0.204/24', mac='00:00:00:00:00:08')
    RailServer = net.addHost('RailServer', cls=Host, ip='192.168.7.104/24', mac='00:00:00:00:00:07')
```

```
    ap1 = net.addAccessPoint('ap1', ssid='ssid-ap1', mode='a', channel='36', position='10,110,0', **kwargs)
    ap2 = net.addAccessPoint('ap2', ssid='ssid-ap2', mode='a', channel='40', position='150,110,0', **kwargs)
```

```
    ap3 = net.addAccessPoint('ap3', ssid='ssid-ap3', mode='a', channel='36', position='10,10,0', **kwargs)
    ap4 = net.addAccessPoint('ap4', ssid='ssid-ap4', mode='a', channel='40', position='150,10,0', **kwargs)
```

```
    net.setPropagationModel(model="logDistance", exp=5)
```

```
    info( "*** Configuring wifi nodes\n" )
    net.configureWifiNodes()
```

```
    info( "*** Creating links\n" )
```

```

s11s22 = {'bw':1000}
net.addLink(s11, s22, cls=TCLink, **s11s22)

s22s33 = {'bw':1000}
net.addLink(s22, s33, cls=TCLink, **s22s33)

CarServers22 = {'bw':1000}
net.addLink(CarServer, s22, cls=TCLink, **CarServers22)

RailServers22 = {'bw':1000}
net.addLink(RailServer, s22, cls=TCLink, **RailServers22)

s11ap1 = {'bw':1000}
net.addLink(s11, ap1, cls=TCLink, **s11ap1)

s11ap3 = {'bw':1000}
net.addLink(s11, ap3, cls=TCLink, **s11ap3)

s33ap2 = {'bw':1000}
net.addLink(s33, ap2, cls=TCLink, **s33ap2)

s33ap4 = {'bw':1000}
net.addLink(s33, ap4, cls=TCLink, **s33ap4)
if '-p' not in args:{
    net.plotGraph(max_x=300, max_y=200)}

if '-s' not in args:
    net.startMobility(time=0)
    net.mobility(Tra1, 'start', time=60, position='10,110,0')
    net.mobility(Tra3, 'start', time=63, position='150,110,0')
    net.mobility(Tra1, 'stop', time=65, position='160,110,0')
    net.mobility(Tra3, 'stop', time=68, position='10,120,0')
    net.stopMobility(time=70)

info("**** Starting network\n")
net.build()
info( '**** Starting controllers\n')
for controller in net.controllers:
    controller.start()

#c1.start()
ap1.start([c1])
ap2.start([c1])
ap3.start([c1])
ap4.start([c1])

net.get('s11').start([c1])
net.get('s22').start([c1])
net.get('s33').start([c1])

info("****Make it self working like iperf and other command\n")

##PING Command
Car1.cmd("xterm -hold -e \"ping 192.168.0.203 \" &")
Tra3.cmd("xterm -hold -e \"ping 192.168.7.102 \" &")

#Tra1.cmd("xterm &")
#xterm -hold -e "python3 "
CarServer.cmd("xterm -hold -e \"sudo python Scapy1.py\" &")
RailServer.cmd("xterm -hold -e \"iperf3 -s \" &")
time.sleep(5)
Tra2.cmd("xterm -hold -e \"iperf3 -c 192.168.7.104 \" &")

##Latency Test
time.sleep(3)
Car2.cmd("xterm -hold -e \"mtr -r -n -c 10 192.168.0.204 -u \" &")

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

info("**** Stopping network\n")
net.stop()
if __name__ == '__main__':
    setLogLevel('info')
    topology(sys.argv)

```

2. S4(5/6)1: - Has single technology, Shared Access Network and Shared Core. Track Parallel to Road

```
#!/usr/bin/python
```

```
'Example for Handover_ Shared Access Network Shared Core'
```

```
import sys
import time
from mininet.log import setLogLevel, info
from mn_wifi.cli import CLI
from mn_wifi.net import Mininet_wifi
from mn_wifi.link import wmediumd
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.log import setLogLevel, info
from mininet.link import TCLink, Intf
from subprocess import call
```

```
from mn_wifi.link import wmediumd, ITSLink
from mn_wifi.wmediumdConnector import interference
from mininet.net import Mininet
```

```
def topology(args):
    "Create a network."
    #net = Mininet_wifi()
    net = Mininet_wifi(topo=None,
                       build=False,
                       ipBase='192.168.0.0/24')
```

```
    info( '*** Adding controller\n' )
    c1=net.addController(name='c1',
                        controller=RemoteController,
                        ip='127.0.0.1',
                        protocol='tcp',
                        port=6653)
```

```
    kwargs = {'protocols':'OpenFlow13','txpower':'41dBm','range': 100 }# 'failMode': 'standalone', 'datapath': 'user'}
```

```
    info( '*** Add switches\n' )
    s3 = net.addSwitch('s3', cls=OVSKernelSwitch, protocols="OpenFlow13")
    s2 = net.addSwitch('s2', cls=OVSKernelSwitch, protocols="OpenFlow13")
    s4 = net.addSwitch('s4', cls=OVSKernelSwitch, protocols="OpenFlow13")
```

```
    info( "*** Creating nodes\n" )
    Tra1_args, Car1_args, Car3_args= dict(), dict(), dict()
    if '-s' in args:
        Tra1_args['position'], Car1_args['position'], Car3_args['position']= '10,10,0', '300,20,0', '10,30,0'
```

```
    Tra1 = net.addStation('Tra1', mac='00:00:00:00:00:01', ip='192.168.7.101/24', position='10,10,0', **Tra1_args)
    Tra2 = net.addStation('Tra2', mac='00:00:00:00:00:03', ip='192.168.7.102/24', position='10,10,0')
    Tra3 = net.addStation('Tra3', mac='00:00:00:00:00:05', ip='192.168.7.103/24', position='300,20,0')
```

```
    Car1 = net.addStation('Car1', mac='00:00:00:00:00:02', ip='192.168.0.201/24', position='300,20,0', **Car1_args)
    Car2 = net.addStation('Car2', mac='00:00:00:00:00:04', ip='192.168.0.202/24', position='300,30,0')
    Car3 = net.addStation('Car3', mac='00:00:00:00:00:06', ip='192.168.0.203/24', position='10,30,0', **Car3_args)
```

```
    info( '*** Add hosts\n' )
    CarServer = net.addHost('CarServer', cls=Host, ip='192.168.0.204/24', mac='00:00:00:00:00:08')
    RailServer = net.addHost('RailServer', cls=Host, ip='192.168.7.104/24', mac='00:00:00:00:00:07')
```

```
    ap1 = net.addAccessPoint('ap1', ssid='ssid-ap1', mode='a', channel='40', position='10,10,0', **kwargs)
    #ap2 = net.addAccessPoint('ap2', ssid='ssid-ap2', mode='a', channel='40', position='50,10,0', protocols="OpenFlow13")
```

```
    ap4 = net.addAccessPoint('ap4', ssid='ssid-ap4', mode='a', channel='36', position='300,20,0', **kwargs)
    net.setPropagationModel(model="logDistance", exp=5)
```

```
    info( "*** Configuring wifi nodes\n" )
    net.configureWifiNodes()
```

```
    info( "*** Creating links\n" )
```

```

s2s3 = {'bw':1000}
net.addLink(s2, s3, cls=TCLink , **s2s3)
s3s4 = {'bw':1000}
net.addLink(s3, s4, cls=TCLink , **s3s4)

s3CarServer = {'bw':1000}
net.addLink(s3, CarServer, cls=TCLink , **s3CarServer)

s3RailServer = {'bw':1000}
net.addLink(s3, RailServer, cls=TCLink , **s3RailServer)

s4ap4 = {'bw':1000}
net.addLink(s4, ap4, cls=TCLink, **s4ap4)

s2ap1 = {'bw':1000}
net.addLink(s2, ap1, cls=TCLink , **s2ap1)

if '-p' not in args:{
    net.plotGraph(max_x=500, min_x=-20, max_y=200, min_y=-20)}

# Strongest Signal First
#net.associationControl('ssf')

if '-s' not in args:
    net.startMobility(time=0)
    net.mobility(Tra1, 'start', time=60, position='10,10,0')
    net.mobility(Car1, 'start', time=62, position='300,20,0')
    net.mobility(Tra1, 'stop', time=67, position='300,30,0')
    net.mobility(Car1, 'stop', time=69, position='10,10,0')
    net.stopMobility(time=70)

info("*** Starting network\n")
net.build()

Tra1 = net.get('Tra1')
Tra2 = net.get('Tra2')
Tra3 = net.get('Tra3')

Car1 = net.get('Car1')
Car2 = net.get('Car2')
Car3 = net.get('Car3')

Tra1.cmd("route add default gw 192.168.7.1 dev Tra1-wlan0")
Tra1.cmd("arp -i Tra1-wlan0 -s 192.168.7.1 08:01:11:01:11:01")

Tra2.cmd("route add default gw 192.168.7.1 dev Tra2-wlan0")
Tra2.cmd("arp -i Tra2-wlan0 -s 192.168.7.1 08:01:11:01:11:01")

Tra3.cmd("route add default gw 192.168.7.1 dev Tra3-wlan0")
Tra3.cmd("arp -i Tra3-wlan0 -s 192.168.7.1 08:01:11:01:11:01")

Car1.cmd("route add default gw 192.168.0.1 dev Car1-wlan0")
Car1.cmd("arp -i Car1-wlan0 -s 192.168.0.1 08:01:22:01:22:01")

Car2.cmd("route add default gw 192.168.0.1 dev Car2-wlan0")
Car2.cmd("arp -i Car2-wlan0 -s 192.168.0.1 08:01:22:01:22:01")

Car3.cmd("route add default gw 192.168.0.1 dev Car3-wlan0")
Car3.cmd("arp -i Car3-wlan0 -s 192.168.0.1 08:01:22:01:22:01")

info( '*** Starting controllers\n')
#Tra1.cmd("ping 192.168.7.103")

for controller in net.controllers:
    controller.start()

#c1.start()
ap1.start([c1])
#ap2.start([c1])
ap4.start([c1])
#ap4.start([c1])

#net.get('s4').start([c1])
net.get('s2').start([c1])

```

```
net.get('s3').start([c1])
net.get('s4').start([c1])
```

```
info("****Make it self working like iperf and other command\n")
```

```
##PING Command
```

```
Car3.cmd("xterm -hold -e \"ping 192.168.0.201 \" &")
```

```
Tra1.cmd("xterm -hold -e \"ping 192.168.7.103 \" &")
```

```
##Latency Test
```

```
Car2.cmd("xterm -hold -e \"mtr -r -n -c 10 192.168.0.204 -u \" &")
```

```
CarServer.cmd("xterm -hold -e \"sudo python Scapy1.py\" &")
```

```
RailServer.cmd("xterm -hold -e \"iperf3 -s \" &")
```

```
time.sleep(5)
```

```
Tra2.cmd("xterm -hold -e \"iperf3 -c 192.168.7.104 \" &")
```

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

```
CLI(net)
```

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

```
net.stop()
```

```
if __name__ == '__main__':
```

```
    setLogLevel('info')
```

```
    topology(sys.argv)
```

3. S4(5/6)4: - Has single technology, Shared Access Network and Shared Core. [Track Perpendicular to Road](#)

```
#!/usr/bin/python
```

```
'Example for Handover_ Shared Access Network Shared Core_Perpendicular'
```

```
import sys
import time
```

```
from mininet.log import setLogLevel, info
from mn_wifi.cli import CLI
from mn_wifi.net import Mininet_wifi
from mn_wifi.link import wmediumd
```

```
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.log import setLogLevel, info
from mininet.link import TCLink, Intf
from subprocess import call
```

```
from mn_wifi.link import wmediumd, ITSLink
from mn_wifi.wmediumdConnector import interference
from mininet.net import Mininet
```

```
def topology(args):
    "Create a network."
    #net = Mininet_wifi()
    net = Mininet_wifi(topo=None,
                       build=False,
                       ipBase='192.168.0.0/24')
```

```
    info( '*** Adding controller\n' )
    c1=net.addController(name='c1',
                        controller=RemoteController,
                        ip='127.0.0.1',
                        protocol='tcp',
                        port=6653)
```

```
    kwargs = {'protocols':'OpenFlow13','txpower':49dBm,'range': 150 }# 'failMode': 'standalone', 'datapath': 'user'}
```

```
    info( '*** Add switches\n' )
    s3 = net.addSwitch('s3', cls=OVSKernelSwitch, protocols="OpenFlow13")
    s2 = net.addSwitch('s2', cls=OVSKernelSwitch, protocols="OpenFlow13")
    s4 = net.addSwitch('s4', cls=OVSKernelSwitch, protocols="OpenFlow13")
```

```
    info("*** Creating nodes\n")
    Tra1_args, Car1_args, Car3_args= dict(), dict(), dict()
    if '-s' in args:
        Tra1_args['position'], Car1_args['position'], Car3_args['position']= '100,180,0', '20,50,0', '120,30,0'
```

```
    Tra1 = net.addStation('Tra1', mac='00:00:00:00:00:01', ip='192.168.7.101/24', position='100,200,0', **Tra1_args)
    Tra2 = net.addStation('Tra2', mac='00:00:00:00:00:03', ip='192.168.7.102/24', position='150,10,0')
    Tra3 = net.addStation('Tra3', mac='00:00:00:00:00:05', ip='192.168.7.103/24', position='270,10,0')
```

```
    Car1 = net.addStation('Car1', mac='00:00:00:00:00:02', ip='192.168.0.201/24', position='10,70,0', **Car1_args)
    Car2 = net.addStation('Car2', mac='00:00:00:00:00:04', ip='192.168.0.202/24', position='10,50,0')
    Car3 = net.addStation('Car3', mac='00:00:00:00:00:06', ip='192.168.0.203/24', position='450,80,0', **Car3_args)
```

```
    info( '*** Add hosts\n' )
    CarServer = net.addHost('CarServer', cls=Host, ip='192.168.0.204/24', mac='00:00:00:00:00:08')
    RailServer = net.addHost('RailServer', cls=Host, ip='192.168.7.104/24', mac='00:00:00:00:00:07')
```

```
    ap1 = net.addAccessPoint('ap1', ssid='ssid-ap1', mode='a', channel='40', position='120,70,0', **kwargs)
    #ap2 = net.addAccessPoint('ap2', ssid='ssid-ap2', mode='a', channel='40', position='50,10,0', protocols="OpenFlow13")
```

```
    ap4 = net.addAccessPoint('ap4', ssid='ssid-ap4', mode='a', channel='36', position='350,70,0', **kwargs)
    #ap4 = net.addAccessPoint('ap4', ssid='ssid-ap4', mode='a', channel='40', position='100,90,0', protocols="OpenFlow13")
    net.setPropagationModel(model="logDistance", exp=5)
```

```
    info("*** Configuring wifi nodes\n")
    net.configureWifiNodes()
```

```

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

s2s3 = {'bw':1000}
net.addLink(s2, s3, cls=TCLink , **s2s3)
s3s4 = {'bw':1000}
net.addLink(s3, s4, cls=TCLink , **s3s4)

s3CarServer = {'bw':1000}
net.addLink(s3, CarServer, cls=TCLink , **s3CarServer)

s3RailServer = {'bw':1000}
net.addLink(s3, RailServer, cls=TCLink , **s3RailServer)

s4ap4 = {'bw':1000}
net.addLink(s4, ap4, cls=TCLink , **s4ap4)

s2ap1 = {'bw':1000}
net.addLink(s2, ap1, cls=TCLink , **s2ap1)

if '-p' not in args:{
    net.plotGraph(max_x=550, min_x=-20, max_y=250, min_y=-20)}

# Strongest Signal First
#net.associationControl('ssf')

if '-s' not in args:
    net.startMobility(time=0)
    net.mobility(Tra1, 'start', time=60, position='100,180,0')
    net.mobility(Car1, 'start', time=62, position='10,70,0')
    net.mobility(Tra1, 'stop', time=67, position='100,10,0')
    net.mobility(Car1, 'stop', time=69, position='450,70,0')
    net.stopMobility(time=70)

info("*** Starting network\n")
net.build()

Tra1 = net.get('Tra1')
Tra2 = net.get('Tra2')
Tra3 = net.get('Tra3')

Car1 = net.get('Car1')
Car2 = net.get('Car2')
Car3 = net.get('Car3')

Tra1.cmd("route add default gw 192.168.7.1 dev Tra1-wlan0")
Tra1.cmd("arp -i Tra1-wlan0 -s 192.168.7.1 08:01:11:01:11:01")

Tra2.cmd("route add default gw 192.168.7.1 dev Tra2-wlan0")
Tra2.cmd("arp -i Tra2-wlan0 -s 192.168.7.1 08:01:11:01:11:01")

Tra3.cmd("route add default gw 192.168.7.1 dev Tra3-wlan0")
Tra3.cmd("arp -i Tra3-wlan0 -s 192.168.7.1 08:01:11:01:11:01")

Car1.cmd("route add default gw 192.168.0.1 dev Car1-wlan0")
Car1.cmd("arp -i Car1-wlan0 -s 192.168.0.1 08:01:22:01:22:01")

Car2.cmd("route add default gw 192.168.0.1 dev Car2-wlan0")
Car2.cmd("arp -i Car2-wlan0 -s 192.168.0.1 08:01:22:01:22:01")

Car3.cmd("route add default gw 192.168.0.1 dev Car3-wlan0")
Car3.cmd("arp -i Car3-wlan0 -s 192.168.0.1 08:01:22:01:22:01")

info('*** Starting controllers\n')
#Tra1.cmd("ping 192.168.7.103")

for controller in net.controllers:
    controller.start()

#c1.start()
ap1.start([c1])
#ap2.start([c1])
ap4.start([c1])
#ap4.start([c1])

```



```
#net.get('s4').start([c1])

net.get('s2').start([c1])

net.get('s3').start([c1])

net.get('s4').start([c1])

info("****Make it self working like iperf and other command\n")

##PING Command

Car3.cmd("xterm -hold -e \"ping 192.168.0.201 \" &")

Tra1.cmd("xterm -hold -e \"ping 192.168.7.103 \" &")

##Latency Test

Car2.cmd("xterm -hold -e \"mtr -r -n -c 10 192.168.0.204 -u \" &")

CarServer.cmd("xterm -hold -e \"sudo python Scapy1.py\" &")

RailServer.cmd("xterm -hold -e \"iperf3 -s \" &")

time.sleep(5)

Tra2.cmd("xterm -hold -e \"iperf3 -c 192.168.7.104 \" &")


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

CLI(net)


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

net.stop()


if __name__ == '__main__':

    setLogLevel('info')

    topology(sys.argv)
```

4. S1(5/6)1: - Has single technology, **Dedicated Access Network for each domain (RAN) and Dedicated Core.** **Track Parallel to Road**

```
#!/usr/bin/python
```

```
'UPDATED Example for Handover for Difference Access and Different Core'
```

```
import sys
import time
```

```
from mininet.log import setLogLevel, info
from mn_wifi.cli import CLI
from mn_wifi.net import Mininet_wifi
```

```
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.log import setLogLevel, info
from mininet.link import TCLink, Intf
from subprocess import call
```

```
def topology(args):
```

```
    "Create a network."
    #net = Mininet_wifi()
```

```
    net = Mininet_wifi( topo=None,
                        build=False,
                        ipBase='192.168.0.0/24')
```

```
    info( '*** Adding controller\n' )
    c1=net.addController(name='c1',
                        controller=RemoteController,
                        ip='127.0.0.1',
                        protocol='tcp',
                        port=6653)
```

```
    kwargs = {'protocols':'OpenFlow13','txpower':'26dBm','range': 50 }# 'failMode': 'standalone', 'datapath': 'user'}
```

```
    info( '*** Add switches\n' )
    s11 = net.addSwitch('s11', cls=OVSKernelSwitch,protocols="OpenFlow13")
    s22 = net.addSwitch('s22', cls=OVSKernelSwitch,protocols="OpenFlow13")
    s33 = net.addSwitch('s33', cls=OVSKernelSwitch,protocols="OpenFlow13")
```

```
    s44 = net.addSwitch('s44', cls=OVSKernelSwitch,protocols="OpenFlow13")
    s55 = net.addSwitch('s55', cls=OVSKernelSwitch,protocols="OpenFlow13")
    s66 = net.addSwitch('s66', cls=OVSKernelSwitch,protocols="OpenFlow13")
```

```
    info("*** Creating nodes\n")
    Tra1_args, Tra3_args, Car1_args = dict(), dict(), dict()
    if '-s' in args:
        Tra1_args['position'], Tra3_args['position'], Car1_args['position'] = '10,10,0', '200,10,0' '10,90,0'
```

```
    Tra1 = net.addStation('Tra1', mac='00:00:00:00:00:01', ip='192.168.7.101/24', position='10,10,0', **Tra1_args)
    Tra2 = net.addStation('Tra2', mac='00:00:00:00:00:03', ip='192.168.7.102/24', position='10,10,0')
    Tra3 = net.addStation('Tra3', mac='00:00:00:00:00:05', ip='192.168.7.103/24', position='200,10,0', **Tra3_args)
```

```
    Car1 = net.addStation('Car1', mac='00:00:00:00:00:02', ip='192.168.0.201/24', position='10,90,0',**Car1_args)
    Car2 = net.addStation('Car2', mac='00:00:00:00:00:04', ip='192.168.0.202/24', position='200,90,0')
    Car3 = net.addStation('Car3', mac='00:00:00:00:00:06', ip='192.168.0.203/24', position='200,90,0')
```

```

info( '*** Add hosts\n')
CarServer = net.addHost('CarServer', cls=Host, ip='192.168.0.204/24', mac='00:00:00:00:00:08')
RailServer = net.addHost('RailServer', cls=Host, ip='192.168.7.104/24', mac='00:00:00:00:00:07')

ap1 = net.addAccessPoint('ap1', ssid='ssid-ap1', mode='a', channel='36', position='10,10,0', **kwargs)
ap2 = net.addAccessPoint('ap2', ssid='ssid-ap2', mode='a', channel='40', position='10,90,0', **kwargs)

ap3 = net.addAccessPoint('ap3', ssid='ssid-ap3', mode='a', channel='36', position='200,10,0', **kwargs)
ap4 = net.addAccessPoint('ap4', ssid='ssid-ap4', mode='a', channel='40', position='200,90,0', **kwargs)

net.setPropagationModel(model="logDistance", exp=5)

info("*** Configuring wifi nodes\n")
net.configureWifiNodes()

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

s11s22 = {'bw':1000}
net.addLink(s11, s22, cls=TCLink , **s11s22)

s22s33 = {'bw':1000}
net.addLink(s22, s33, cls=TCLink , **s22s33)

s44s55 = {'bw':1000}
net.addLink(s44, s55, cls=TCLink , **s44s55)

s55s66 = {'bw':1000}
net.addLink(s55, s66, cls=TCLink , **s55s66)

CarServers22 = {'bw':1000}
net.addLink(CarServer, s22, cls=TCLink , **CarServers22)

RailServers55 = {'bw':1000}
net.addLink(RailServer, s55, cls=TCLink , **RailServers55)

s44ap1 = {'bw':1000}
net.addLink(s44, ap1, cls=TCLink , **s44ap1)

s11ap2 = {'bw':1000}
net.addLink(s11, ap2, cls=TCLink , **s11ap2)

s66ap3 = {'bw':1000}
net.addLink(s66, ap3, cls=TCLink , **s66ap3)

s33ap4 = {'bw':1000}
net.addLink(s33, ap4, cls=TCLink , **s33ap4)

if '-p' not in args:{
    net.plotGraph(max_x=300, max_y=200)}

if '-s' not in args:
    net.startMobility(time=0)
    net.mobility(Tra1, 'start', time=60, position='10,10,0')
    net.mobility(Tra3, 'start', time=63, position='200,10,0')
    net.mobility(Tra1, 'stop', time=70, position='200,10,0')
    net.mobility(Tra3, 'stop', time=73, position='10,20,0')
    net.stopMobility(time=75)

info("*** Starting network\n")
net.build()
info( '*** Starting controllers\n')
for controller in net.controllers:
    controller.start()

```

```
#c1.start()
ap1.start([c1])
ap2.start([c1])
ap3.start([c1])
ap4.start([c1])
```

```
net.get('s11').start([c1])
net.get('s22').start([c1])
net.get('s33').start([c1])
net.get('s44').start([c1])
net.get('s55').start([c1])
net.get('s66').start([c1])
```

```
info("****Make it self working like iperf and other command\n")
```

```
##PING Command
```

```
Car1.cmd("xterm -hold -e \"ping 192.168.0.203 \" &")
```

```
Tra3.cmd("xterm -hold -e \"ping 192.168.7.102 \" &")
```

```
#Tra1.cmd("xterm &")
```

```
#xterm -hold -e "python3 "
```

```
CarServer.cmd("xterm -hold -e \"sudo python Scapy1.py\" &")
```

```
RailServer.cmd("xterm -hold -e \"iperf3 -s \" &")
```

```
time.sleep(5)
```

```
Tra2.cmd("xterm -hold -e \"iperf3 -c 192.168.7.104 \" &")
```

```
##Latency Test
```

```
time.sleep(5)
```

```
Car2.cmd("xterm -hold -e \"mtr -r -n -c 10 192.168.0.204 -u \" &")
```

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

```
CLI(net)
```

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

```
net.stop()
```

```
if __name__ == '__main__':
```

```
    setLogLevel('info')
```

```
    topology(sys.argv)
```

5. S1(5/6)4: - Has single technology, **Dedicated Access Network** for each domain (RAN) and **Dedicated Core**. **Track Perpendicular to Road**

```
#!/usr/bin/python
```

```
'UPDATED Example for Handover for Difference Access and Different Core_Perpendicular'
```

```
import sys
import time
```

```
from mininet.log import setLogLevel, info
from mn_wifi.cli import CLI
from mn_wifi.net import Mininet_wifi
```

```
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.log import setLogLevel, info
from mininet.link import TCLink, Intf
from subprocess import call
```

```
def topology(args):
```

```
    "Create a network."
    #net = Mininet_wifi()
```

```
    net = Mininet_wifi( topo=None,
                        build=False,
                        ipBase='192.168.0.0/24')
```

```
    info( '*** Adding controller\n' )
    c1=net.addController(name='c1',
                        controller=RemoteController,
                        ip='127.0.0.1',
                        protocol='tcp',
                        port=6653)
```

```
    kwargs = {'protocols':'OpenFlow13','txpower':'49dBm','range': 100 }# 'failMode': 'standalone', 'datapath': 'user'}
```

```
    info( '*** Add switches\n' )
    s11 = net.addSwitch('s11', cls=OVSKernelSwitch,protocols="OpenFlow13")
    s22 = net.addSwitch('s22', cls=OVSKernelSwitch,protocols="OpenFlow13")
    s33 = net.addSwitch('s33', cls=OVSKernelSwitch,protocols="OpenFlow13")
```

```
    s44 = net.addSwitch('s44', cls=OVSKernelSwitch,protocols="OpenFlow13")
    s55 = net.addSwitch('s55', cls=OVSKernelSwitch,protocols="OpenFlow13")
    s66 = net.addSwitch('s66', cls=OVSKernelSwitch,protocols="OpenFlow13")
```

```
    info( "*** Creating nodes\n" )
    Tra1_args, Tra3_args, Car1_args = dict(), dict(), dict()
    if '-s' in args:
        Tra1_args['position'], Tra3_args['position'], Car1_args['position'] = '200,450,0', '250.10.0' '30,220,0'
```

```
    Tra1 = net.addStation('Tra1', mac='00:00:00:00:00:01', ip='192.168.7.101/24', position='200,450,0', **Tra1_args)
    Tra2 = net.addStation('Tra2', mac='00:00:00:00:00:03', ip='192.168.7.102/24', position='200,430,0')
    Tra3 = net.addStation('Tra3', mac='00:00:00:00:00:05', ip='192.168.7.103/24', position='250,10,0', **Tra3_args)
```

```
    Car1 = net.addStation('Car1', mac='00:00:00:00:00:02', ip='192.168.0.201/24', position='30,220,0',**Car1_args)
    Car2 = net.addStation('Car2', mac='00:00:00:00:00:04', ip='192.168.0.202/24', position='30,230,0')
    Car3 = net.addStation('Car3', mac='00:00:00:00:00:06', ip='192.168.0.203/24', position='360,240,0')
```

```
    info( '*** Add hosts\n' )
    CarServer = net.addHost('CarServer', cls=Host, ip='192.168.0.204/24', mac='00:00:00:00:00:08')
    RailServer = net.addHost('RailServer', cls=Host, ip='192.168.7.104/24', mac='00:00:00:00:00:07')
```

```
    ap1 = net.addAccessPoint('ap1', ssid='ssid-ap1', mode='a', channel='36', position='100,220,0', **kwargs)
    ap2 = net.addAccessPoint('ap2', ssid='ssid-ap2', mode='a', channel='40', position='300,220,0', **kwargs)
```

```
    ap3 = net.addAccessPoint('ap3', ssid='ssid-ap3', mode='a', channel='36', position='200,400,0', **kwargs)
    ap4 = net.addAccessPoint('ap4', ssid='ssid-ap4', mode='a', channel='40', position='200,40,0', **kwargs)
```

```

net.setPropagationModel(model="logDistance", exp=5)

info("*** Configuring wifi nodes\n")
net.configureWifiNodes()

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

s11s22 = {'bw':1000}
net.addLink(s11, s22, cls=TCLink, **s11s22)

s22s33 = {'bw':1000}
net.addLink(s22, s33, cls=TCLink, **s22s33)

s44s55 = {'bw':1000}
net.addLink(s44, s55, cls=TCLink, **s44s55)

s55s66 = {'bw':1000}
net.addLink(s55, s66, cls=TCLink, **s55s66)

CarServers22 = {'bw':1000}
net.addLink(CarServer, s22, cls=TCLink, **CarServers22)

RailServers55 = {'bw':1000}
net.addLink(RailServer, s55, cls=TCLink, **RailServers55)

s44ap3 = {'bw':1000}
net.addLink(s44, ap3, cls=TCLink, **s44ap3)

s11ap1 = {'bw':1000}
net.addLink(s11, ap1, cls=TCLink, **s11ap1)

s66ap4 = {'bw':1000}
net.addLink(s66, ap4, cls=TCLink, **s66ap4)

s33ap2 = {'bw':1000}
net.addLink(s33, ap2, cls=TCLink, **s33ap2)

if '-p' not in args:
    net.plotGraph(max_x=500,min_x=-50, max_y=500, min_y=-60)}

if '-s' not in args:
    net.startMobility(time=0)
    net.mobility(Tra1, 'start', time=60, position='200,450,0')
    net.mobility(Car1, 'start', time=63, position='30,200,0')
    net.mobility(Tra1, 'stop', time=70, position='200,10,0')
    net.mobility(Car1, 'stop', time=73, position='360,200,0')
    net.stopMobility(time=75)

info("*** Starting network\n")
net.build()
info('*** Starting controllers\n')
for controller in net.controllers:
    controller.start()

#c1.start()
ap1.start([c1])
ap2.start([c1])
ap3.start([c1])
ap4.start([c1])

net.get('s11').start([c1])
net.get('s22').start([c1])
net.get('s33').start([c1])
net.get('s44').start([c1])
net.get('s55').start([c1])
net.get('s66').start([c1])

info("****Make it self working like iperf and other command\n")

##PING Command
Car1.cmd("xterm -hold -e \"ping 192.168.0.203 \" &")
Tra1.cmd("xterm -hold -e \"ping 192.168.7.103 \" &")

```

```
#Tra1.cmd("xterm &")
#xterm -hold -e "python3 "
CarServer.cmd("xterm -hold -e \"sudo python Scapy1.py\" &")
RailServer.cmd("xterm -hold -e \"iperf3 -s \" &")
time.sleep(5)
Tra2.cmd("xterm -hold -e \"iperf3 -c 192.168.7.104 \" &")

##Latency Test
time.sleep(5)
Car2.cmd("xterm -hold -e \"mtr -r -n -c 10 192.168.0.204 -u \" &")

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

info("*** Stopping network\n")
net.stop()

if __name__ == '__main__':
    setLogLevel('info')
    topology(sys.argv)
```