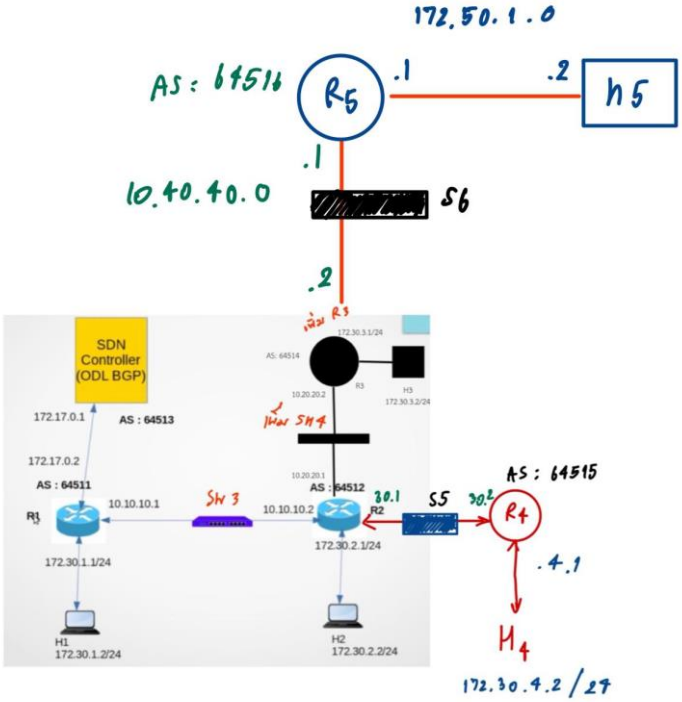
เพิ่ม R4 กับ R5



1.Topo1.py

```
test@test: ~/Desktop/ODL_week9/odlbgp/add_r4-r5
test@test:~/Desktop/ODL_week9/odlbgp/add_r4-r5$ sudo python3 topo1.py
*** Adding switch
r1: kwargs {'ip': '172.30.1.1/24'}
r1: update resources {}
r2: kwargs {'ip': '172.30.2.1/24'}
r2: update resources {}
r3: kwargs {'ip': '172.30.3.1/24'}
r3: update resources {}
r4: kwargs {'ip': '172.30.4.1/24'}
r4: update resources {}
r5: kwargs {'ip': '172.30.5.1/24'}
r5: update resources {}
h1: kwargs {'ip': '172.30.1.2/24', 'defaultRoute': 'via 172.30.1.1'}
h1: update resources {}
h2: kwargs {'ip': '172.30.2.2/24', 'defaultRoute': 'via 172.30.2.1'}
h2: update resources {}
h3: kwargs {'ip': '172.30.3.2/24', 'defaultRoute': 'via 172.30.3.1'}
h3: update resources {}
h4: kwargs {'ip': '172.30.4.2/24', 'defaultRoute': 'via 172.30.4.1'}
h4: update resources {}
h5: kwargs {'ip': '172.30.5.2/24', 'defaultRoute': 'via 172.30.5.1'}
h5: update resources {}
*** Creating links
*** Starting network
*** Configuring hosts
r1 r2 r3 r4 r5 h1 h2 h3 h4 h5
*** Starting controller

*** Starting 4 switches
s3 s4 s5 s6 ...
*** Running CLI
*** Starting CLI:
```

2. links

```
containernet> links
h1-eth0<->r1-eth0 (OK OK)
h2-eth0<->r2-eth0 (OK OK)
h3-eth0<->r3-eth0 (OK OK)
h4-eth0<->r4-eth0 (OK OK)
h5-eth0<->r5-eth0 (OK OK)
r1-eth1<->s3-eth1 (OK OK)
r2-eth1<->s3-eth2 (OK OK)
r2-eth2<->s4-eth1 (OK OK)
r2-eth3<->s5-eth1 (OK OK)
r3-eth1<->s4-eth2 (OK OK)
r4-eth1<->s5-eth2 (OK OK)
r3-eth2<->s6-eth1 (OK OK)
r5-eth1<->s6-eth2 (OK OK)
containernet>
```

3. ping

```
containernet> h1 ping h5
PING 172.30.5.2 (172.30.5.2) 56(84) bytes of data.
64 bytes from 172.30.5.2: icmp_seq=1 ttl=60 time=0.423 ms
64 bytes from 172.30.5.2: icmp_seq=2 ttl=60 time=0.230 ms
^C
--- 172.30.5.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1019ms
rtt min/avg/max/mdev = 0.230/0.326/0.423/0.098 ms
containernet>
```

```
containernet> h1 ping h4
PING 172.30.4.2 (172.30.4.2) 56(84) bytes of data.
64 bytes from 172.30.4.2: icmp_seq=1 ttl=61 time=0.299 ms
64 bytes from 172.30.4.2: icmp_seq=2 ttl=61 time=0.184 ms
^C
--- 172.30.4.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1007ms
rtt min/avg/max/mdev = 0.184/0.241/0.299/0.059 ms
containernet>
```

4. traceroute

```
containernet> h1 traceroute h4
traceroute to 172.30.4.2 (172.30.4.2), 64 hops max
  1   172.30.1.1  0.006ms  0.003ms  0.004ms
  2   10.10.10.2  0.004ms  0.003ms  0.003ms
  3   10.30.30.2  0.343ms  0.004ms  0.003ms
  4   172.30.4.2  0.003ms  0.003ms  0.003ms
containernet>
```

```
containernet> h1 traceroute h5
traceroute to 172.30.5.2 (172.30.5.2), 64 hops max
  1   172.30.1.1  0.003ms  0.002ms  0.002ms
  2   10.10.10.2  0.002ms  0.002ms  0.001ms
  3   10.20.20.2  0.287ms  0.002ms  0.001ms
  4   10.40.40.1  0.720ms  0.002ms  0.001ms
  5   172.30.5.2  0.001ms  0.001ms  0.000ms
containernet>
```

Code

Topo1.py

Open ▾ topo1.py ~/Desktop/ODL_week9/odlbgp/add_r4-r5 Save

```
from mininet.net import Containernet
from mininet.node import RemoteController, Docker, OVSSwitch
from mininet.cli import CLI
from mininet.log import setLogLevel, info
from mininet.link import TCLink, Link

# topo diagram
# h1----r1---s3---r2-----h2


def topology():

    "Create a network with some docker containers acting as hosts."

    net = Containernet(controller=RemoteController)

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

    #add r
    r1 = net.addDocker('r1', ip='172.30.1.1/24', dimage="knet/urouter:1.4")
    r2 = net.addDocker('r2', ip='172.30.2.1/24', dimage="knet/urouter:1.4")
    r3 = net.addDocker('r3', ip='172.30.3.1/24', dimage="knet/urouter:1.4")
    r4 = net.addDocker('r4', ip='172.30.4.1/24', dimage="knet/urouter:1.4")
    r5 = net.addDocker('r5', ip='172.30.5.1/24', dimage="knet/urouter:1.4")

    #add h
    h1 = net.addDocker('h1', ip='172.30.1.2/24', defaultRoute='via 172.30.1.1' ,dimage="knet/host-ubuntu:1-2")
    h2 = net.addDocker('h2', ip='172.30.2.2/24', defaultRoute='via 172.30.2.1' ,dimage="knet/host-ubuntu:1-2")
    h3 = net.addDocker('h3', ip='172.30.3.2/24', defaultRoute='via 172.30.3.1' ,dimage="knet/host-ubuntu:1-2")
    h4 = net.addDocker('h4', ip='172.30.4.2/24', defaultRoute='via 172.30.4.1' ,dimage="knet/host-ubuntu:1-2")
    h5 = net.addDocker('h5', ip='172.30.5.2/24', defaultRoute='via 172.30.5.1' ,dimage="knet/host-ubuntu:1-2")

    #add sw
    s3 = net.addSwitch('s3', failMode='standalone')
    s4 = net.addSwitch('s4', failMode='standalone')
    s5 = net.addSwitch('s5', failMode='standalone')
    s6 = net.addSwitch('s6', failMode='standalone')
```

Open

topo1.py
~/Desktop/ODL_week9/odlbgp/add_r4-r5

```
    s4 = net.addSwitch('s4', failMode='standalone')
    s5 = net.addSwitch('s5', failMode='standalone')
    s6 = net.addSwitch('s6', failMode='standalone')

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

    #add h - r
    net.addLink(h1, r1)
    net.addLink(h2, r2)
    net.addLink(h3, r3)
    net.addLink(h4, r4)
    net.addLink(h5, r5)

    #add r - sw
    net.addLink(r1, s3, params1={"ip": "10.10.10.1/24"})
    net.addLink(r2, s3, params1={"ip": "10.10.10.2/24"})
    net.addLink(r2, s4, params1={"ip": "10.20.20.1/24"})
    net.addLink(r2, s5, params1={"ip": "10.30.30.1/24"})
    net.addLink(r3, s4, params1={"ip": "10.20.20.2/24"})
    net.addLink(r4, s5, params1={"ip": "10.30.30.2/24"})
    net.addLink(r3, s6, params1={"ip": "10.40.40.2/24"})
    net.addLink(r5, s6, params1={"ip": "10.40.40.1/24"})


    info('*** Starting network\n')
    net.start()

    #copy the bird config files
    s3.cmd("sudo docker cp r1.conf mn.r1:/etc/bird.conf")
    s3.cmd("sudo docker cp r2.conf mn.r2:/etc/bird.conf")
    s3.cmd("sudo docker cp r3.conf mn.r3:/etc/bird.conf")
    s3.cmd("sudo docker cp r4.conf mn.r4:/etc/bird.conf")
    s3.cmd("sudo docker cp r5.conf mn.r5:/etc/bird.conf")

    s4.cmd("sudo docker cp r1.conf mn.r1:/etc/bird.conf")
    s4.cmd("sudo docker cp r2.conf mn.r2:/etc/bird.conf")
    s4.cmd("sudo docker cp r3.conf mn.r3:/etc/bird.conf")
    s4.cmd("sudo docker cp r4.conf mn.r4:/etc/bird.conf")
    s4.cmd("sudo docker cp r5.conf mn.r5:/etc/bird.conf")

    s5.cmd("sudo docker cp r1.conf mn.r1:/etc/bird.conf")
    s5.cmd("sudo docker cp r2.conf mn.r2:/etc/bird.conf")
    s5.cmd("sudo docker cp r3.conf mn.r3:/etc/bird.conf")
    s5.cmd("sudo docker cp r4.conf mn.r4:/etc/bird.conf")
    s5.cmd("sudo docker cp r5.conf mn.r5:/etc/bird.conf")

    #add r
    r1.cmd("bird -c /etc/bird.conf")
    r2.cmd("bird -c /etc/bird.conf")
    r3.cmd("bird -c /etc/bird.conf")
    r4.cmd("bird -c /etc/bird.conf")
    r5.cmd("bird -c /etc/bird.conf")


    info('*** Running CLI\n')
    CLI(net)
    info('*** Stopping network')
    net.stop()

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

R1.conf

```
log "/var/log/bird.log"  all;
debug protocols all

router id 10.10.10.1;
protocol direct {
    interface "*";
}

protocol kernel {
    learn;
    scan time 20;
    export all;
    import all;
}


protocol device {
    scan time 10;
}



#BGP Configuration

protocol bgp R2{
        export all;
        import all;
        local as 64511;
        neighbor 10.10.10.2 as 64512;
}

protocol bgp R3{
        export all;
        import all;
        local as 64511;
        neighbor 10.20.20.2 as 64514;
}

protocol bgp R4{
         export all;
         import all;
         local as 64511;
         neighbor 10.30.30.2 as 64515;
}

protocol bgp R5{
         export all;
         import all;
         local as 64511;
         neighbor 10.40.40.1 as 64516;
}

protocol bgp ODl{
         export all;
         import all;
         local as 64511;
         neighbor 172.17.0.1 as 64513;
}
```

R2.conf

Open

```
log "/var/log/bird.log"  all;
debug protocols all

router id 10.10.10.2;
protocol direct {
    interface "*";
}


protocol kernel {
    learn;
    scan time 20;
    export all;
    import all;
}


protocol device {
    scan time 10;
}



#BGP Configuration

protocol bgp R1{
        export all;
        import all;
        local as 64512;
        neighbor 10.10.10.1 as 64511;
}

protocol bgp R3{
        export all;
        import all;
        local as 64512;
        neighbor 10.20.20.2 as 64514;
}

protocol bgp R4{
        export all;
        import all;
        local as 64512;
        neighbor 10.30.30.2 as 64515;
}

protocol bgp R5{
        export all;
        import all;
        local as 64512;
        neighbor 10.40.40.1 as 64516;
}

```

R3.conf

Open

```
log "/var/log/bird.log"  all;
debug protocols all

router id 10.20.20.2;
protocol direct {
    interface "*";
}


protocol kernel {
    learn;
    scan time 20;
    export all;
    import all;
}


protocol device {
    scan time 10;
}



#BGP Configuration

protocol bgp R2{
        export all;
        import all;
        local as 64514;
        neighbor 10.20.20.1 as 64512;
}

protocol bgp R1{
        export all;
        import all;
        local as 64514;
        neighbor 10.10.10.1 as 64511;
}

protocol bgp R4{
        export all;
        import all;
        local as 64514;
        neighbor 10.30.30.2 as 64515;
}

protocol bgp R5{
        export all;
        import all;
        local as 64514;
        neighbor 10.40.40.1 as 64516;
}
```

R4.conf

Open

```
log "/var/log/bird.log"  all;
debug protocols all

router id 10.30.30.2;
protocol direct {
    interface "*";
}


protocol kernel {
    learn;
    scan time 20;
    export all;
    import all;
}


protocol device {
    scan time 10;
}



#BGP Configuration

protocol bgp R2{
        export all;
        import all;
        local as 64515;
        neighbor 10.30.30.1 as 64512;
}

protocol bgp R1{
        export all;
        import all;
        local as 64515;
        neighbor 10.10.10.1 as 64511;
}

protocol bgp R3{
        export all;
        import all;
        local as 64515;
        neighbor 10.20.20.2 as 64514;
}

protocol bgp R5{
        export all;
        import all;
        local as 64515;
        neighbor 10.40.40.1 as 64516;
}
```

R5.conf

Open

```
log "/var/log/bird.log"  all;
debug protocols all

router id 10.40.40.1;
protocol direct {
    interface "*";
}


protocol kernel {
    learn;
    scan time 20;
    export all;
    import all;
}


protocol device {
    scan time 10;
}



#BGP Configuration

protocol bgp R2{
        export all;
        import all;
        local as 64516;
        neighbor 10.20.20.1 as 64512;
}

protocol bgp R1{
        export all;
        import all;
        local as 64516;
        neighbor 10.10.10.1 as 64511;
}

protocol bgp R3{
        export all;
        import all;
        local as 64516;
        neighbor 10.40.40.2 as 64514;
}

protocol bgp R4{
        export all;
        import all;
        local as 64516;
        neighbor 10.30.30.2 as 64515;
}
```