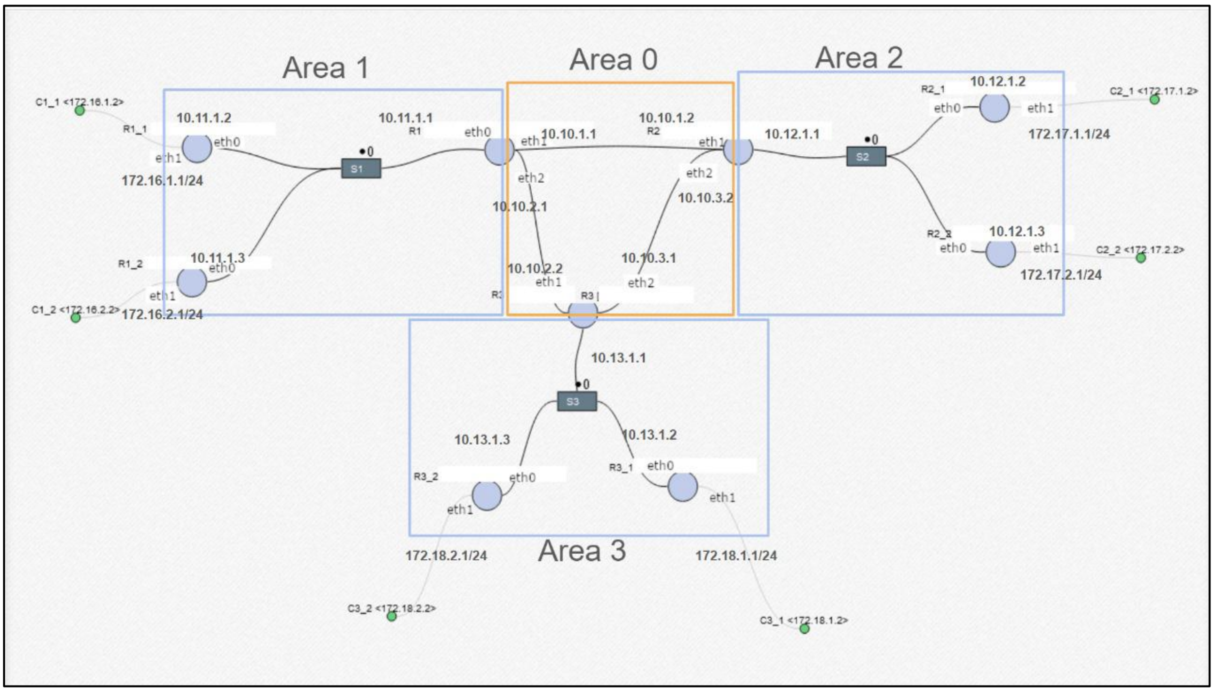
Tugas Intra-domain Routing dengan OSPF Single Area dan Multi Area – Mininet

Dani Adrian
225150201111009

1. **Topologi Jaringan**



Topologi ini terdiri dari:

- 9 Router (R1, R2, R3, R1_1, R1_2, R2_1, R2_2, R3_1, R3_2)
- 3 Switch (S1, S2, S3)
- 6 Host (C1_1, C1_2, C2_1, C2_2, C3_1, C3_2)

2. **Konfigurasi**

FRR Config Single

- R1 :

```
frr version 8.5.4
frr defaults traditional
hostname R1
service integrated-vtysh-config
!
interface R1-eth0
 ip address 10.11.1.1/24
exit
!
interface R1-eth1
 ip address 10.10.1.1/24
exit
!
interface R1-eth2
 ip address 10.10.2.1/24
exit
```

```
!
router ospf
 ospf router-id 1.1.1.1
 network 10.11.1.0/24 area 0
 network 10.10.1.0/24 area 0
 network 10.10.2.0/24 area 0
exit
!
line vty
```

- R1_1 :

```
frr version 8.5.4
frr defaults traditional
hostname R1_1
service integrated-vtysh-config
!
interface R1_1-eth0
 ip address 10.11.1.2/24
exit
!
interface R1_1-eth1
 ip address 172.16.1.1/24
exit
!
router ospf
 ospf router-id 1.1.1.2
 network 10.11.1.0/24 area 0
 network 172.16.1.0/24 area 0
exit
!
line vty
```

- R1_2 :

```
frr version 8.5.4
frr defaults traditional
hostname R1_2
service integrated-vtysh-config
!
interface R1_2-eth0
 ip address 10.11.1.3/24
exit
!
interface R1_2-eth1
 ip address 172.16.2.1/24
exit
```

```
!
router ospf
 ospf router-id 1.1.1.3
 network 10.11.1.0/24 area 0
 network 172.16.2.0/24 area 0
exit
!
line vty
```

- R2 :

```
frr version 8.5.4
frr defaults traditional
hostname R2
service integrated-vtysh-config
!
interface R2-eth0
 ip address 10.12.1.1/24
exit
!
interface R2-eth1
 ip address 10.10.1.2/24
exit
!
interface R2-eth2
 ip address 10.10.3.2/24
exit
!
router ospf
 ospf router-id 2.2.2.1
 network 10.12.1.0/24 area 0
 network 10.10.1.0/24 area 0
 network 10.10.3.0/24 area 0
exit
!
line vty
```

- R2_1 :

```
frr version 8.5.4
frr defaults traditional
hostname R2_1
service integrated-vtysh-config
!
interface R2_1-eth0
 ip address 10.12.1.2/24
exit
```

```
!
interface R2_1-eth1
 ip address 172.17.1.1/24
exit
!
router ospf
 ospf router-id 2.2.2.2
 network 10.12.1.0/24 area 0
 network 172.17.1.0/24 area 0
exit
!
line vty
```

- R2_2 :

```
frr version 8.5.4
frr defaults traditional
hostname R2_2
service integrated-vtysh-config
!
interface R2_2-eth0
 ip address 10.12.1.3/24
exit
!
interface R2_2-eth1
 ip address 172.17.2.1/24
exit
!
router ospf
 ospf router-id 2.2.2.3
 network 10.12.1.0/24 area 0
 network 172.17.2.0/24 area 0
exit
!
line vty
```

- R3 :

```
frr version 8.5.4
frr defaults traditional
frr version 8.5.4
frr defaults traditional
hostname R3
service integrated-vtysh-config
!
interface R3-eth0
 ip address 10.13.1.1/24
```

```
exit
!
interface R3-eth1
 ip address 10.10.2.2/24
exit
!
interface R3-eth2
 ip address 10.10.3.1/24
exit
!
router ospf
 ospf router-id 3.3.3.1
 network 10.13.1.0/24 area 0
 network 10.10.2.0/24 area 0
 network 10.10.3.0/24 area 0
exit
!
line vty
```

- R3_1 :

```
frr version 8.5.4
frr defaults traditional
hostname R3_1
service integrated-vtysh-config
!
interface R3_1-eth0
 ip address 10.13.1.2/24
exit
!
interface R3_1-eth1
 ip address 172.18.1.1/24
exit
!
router ospf
 ospf router-id 3.3.3.2
 network 10.13.1.0/24 area 0
 network 172.18.1.0/24 area 0
exit
!
line vty
```

- R3_2 :

```
frr version 8.5.4
frr defaults traditional
hostname R3_2
```

```
service integrated-vtysh-config
!
interface R3_2-eth0
 ip address 10.13.1.3/24
exit
!
interface R3_2-eth1
 ip address 172.18.2.1/24
exit
!
router ospf
 ospf router-id 3.3.3.3
 network 10.13.1.0/24 area 0
 network 172.18.2.0/24 area 0
exit
!
line vty
```

FRR Config Multi Area

- R1 :

```
frr version 8.5.4
frr defaults traditional
hostname R1
service integrated-vtysh-config
!
interface R1-eth0
 ip ospf network broadcast
 ip address 10.11.1.1/24
exit
!
interface R1-eth1
 ip address 10.10.1.1/24
exit
!
interface R1-eth2
 ip address 10.10.2.1/24
exit
!
router ospf
 ospf router-id 1.1.1.1
 network 10.11.1.0/24 area 1
 network 10.10.1.0/24 area 0
 network 10.10.2.0/24 area 0
exit
!
```

```
line vty
```

- R1_1 :

```
frr version 8.5.4
frr defaults traditional
hostname R1_1
service integrated-vtysh-config
!
interface R1_1-eth0
 ip address 10.11.1.2/24
exit
!
interface R1_1-eth1
 ip address 172.16.1.1/24
exit
!
router ospf
 ospf router-id 1.1.1.2
 network 10.11.1.0/24 area 1
 network 172.16.1.0/24 area 1
exit
!
line vty
```

- R1_2 :

```
frr version 8.5.4
frr defaults traditional
hostname R1_2
service integrated-vtysh-config
!
interface R1_2-eth0
 ip address 10.11.1.3/24
exit
!
interface R1_2-eth1
 ip address 172.16.2.1/24
exit
!
router ospf
 ospf router-id 1.1.1.3
 network 10.11.1.0/24 area 1
 network 172.16.2.0/24 area 1
exit
!
line vty
```

- R2 :

```
frr version 8.5.4
frr defaults traditional
hostname R2
service integrated-vtysh-config
!
interface R2-eth0
 ip address 10.12.1.1/24
exit
!
interface R2-eth1
 ip address 10.10.1.2/24
exit
!
interface R2-eth2
 ip address 10.10.3.2/24
exit
!
router ospf
 ospf router-id 2.2.2.1
 network 10.12.1.0/24 area 2
 network 10.10.1.0/24 area 0
 network 10.10.3.0/24 area 0
exit
!
line vty
```

- R2_1 :

```
frr version 8.5.4
frr defaults traditional
hostname R2_1
service integrated-vtysh-config
!
interface R2_1-eth0
 ip address 10.12.1.2/24
exit
!
interface R2_1-eth1
 ip address 172.17.1.1/24
exit
!
router ospf
 ospf router-id 2.2.2.2
 network 10.12.1.0/24 area 2
 network 172.17.1.0/24 area 2
exit
!
line vty
```

- R2_2 :

```
frr version 8.5.4
frr defaults traditional
hostname R2_2
service integrated-vtysh-config
!
interface R2_2-eth0
 ip address 10.12.1.3/24
exit
!
interface R2_2-eth1
 ip address 172.17.2.1/24
exit
!
router ospf
 ospf router-id 2.2.2.3
 network 10.12.1.0/24 area 2
 network 172.17.2.0/24 area 2
exit
!
line vty
```

- R3 :

```
frr version 8.5.4
frr defaults traditional
hostname R3
service integrated-vtysh-config
!
interface R3-eth0
 ip address 10.13.1.1/24
exit
!
interface R3-eth1
 ip address 10.10.2.2/24
exit
!
interface R3-eth2
 ip address 10.10.3.1/24
exit
!
```

```
router ospf
 ospf router-id 3.3.3.1
 network 10.13.1.0/24 area 3
 network 10.10.2.0/24 area 0
 network 10.10.3.0/24 area 0
exit
!
line vty
```

- R3_1 :

```
frr version 8.5.4
frr defaults traditional
hostname R3_1
service integrated-vtysh-config
!
interface R3_1-eth0
 ip address 10.13.1.2/24
exit
!
interface R3_1-eth1
 ip address 172.18.1.1/24
exit
!
router ospf
 ospf router-id 3.3.3.2
 network 10.13.1.0/24 area 3
 network 172.18.1.0/24 area 3
exit
!
line vty
```

- R3_2 :

```
frr version 8.5.4
frr defaults traditional
hostname R3_2
service integrated-vtysh-config
!
interface R3_2-eth0
 ip address 10.13.1.3/24
exit
!
interface R3_2-eth1
 ip address 172.18.2.1/24
exit
!
```

```
router ospf
 ospf router-id 3.3.3.3
 network 10.13.1.0/24 area 3
 network 172.18.2.0/24 area 3
exit
!
line vty
```

3. Implementasi

Script Mininet :

ospf-lab.py

```
#!/usr/bin/env python3

from mininet.node import CPULimitedHost, Host, Node
from mininet.node import OVSKernelSwitch, UserSwitch
from mininet.log import setLogLevel, info
from mininet.link import TCLink, Intf
from subprocess import call
import shutil
import time
from pathlib import Path
from mininet.topo import Topo
from mininet.net import Mininet
from mininet.cli import CLI
from mininet.nodelib import LinuxBridge
import argparse

class LinuxRouter( Node ):
     def config( self, **params ):
          super( LinuxRouter, self).config( **params )
          self.cmd( 'sysctl -w net.ipv4.ip_forward=1' )
          self.cmd('/usr/lib/frr/zebra -A 127.0.0.1 -s 
90000000 -f /etc/frr/frr.conf -d')
          self.cmd('/usr/lib/frr/staticd -A 127.0.0.1 -f 
/etc/frr/frr.conf -d')
          self.cmd('/usr/lib/frr/ospfd -A 127.0.0.1 -f 
/etc/frr/frr.conf -d')
          self.cmd('/usr/lib/frr/bgpd -A 127.0.0.1 -f 
/etc/frr/frr.conf -d')

          # region
          # self.cmd( 'sysctl -w 
net.ipv6.conf.all.forwarding=1' )
          # self.cmd('/usr/lib/frr/pimd -A 127.0.0.1 -f 
/etc/frr/frr.conf -d')
```

```
          # self.cmd('/usr/lib/frr/pim6d -A ::1 -f
/etc/frr/frr.conf -d')
          # self.cmd('/usr/lib/frr/isisd -A 127.0.0.1 -f
/etc/frr/frr.conf -d')
          # self.cmd('/usr/lib/frr/ospf6d -A ::1 -f
/etc/frr/frr.conf -d')
          # endregion

     def terminate( self ):
          self.cmd( 'killall zebra staticd ospfd ospf6d
bgpd pathd pimd pim6d ldpd isisd nhrpd vrrpd fabricd' )
          super( LinuxRouter, self ).terminate()

     def start (self):
          return

class OSPFLab(Topo):

     def generate_config(self, router_name, path):
          """ Generate an empty config for each
router.\n
               path: the path of router configs
directory
          """
          router = {"name":router_name}
          path = path % router
          #print(path)
          #config template directory path
          template_path = Path("Template/router")
          Path(path).mkdir(exist_ok=True, parents=True)

          #copy files from the config template folder
          for file in template_path.iterdir():
               shutil.copy(file, path)

          #modify hostname
          self.replace_hostname(path+"/frr.conf",
"dummy", router_name)
          self.replace_hostname(path+"/vtysh.conf",
"dummy", router_name)

          self.add_ospf_configuration(path+"/frr.conf",
router_name)

          return
     def replace_hostname(self, filepath, toReplace, 
replacement):
          """ Replace hostname in a router config \n
               filepath: path to the config file\n
               toReplace: the hostname to replace\n
               replacement: the new hostname\n
          """
          with open(filepath, 'r') as f:
               content = f.readlines()
               for linenum in range (len(content)):
                    if (content[linenum] == "hostname 
"+toReplace+"\n"):
                         content[linenum] = "hostname "+
replacement+"\n"
          with open(filepath, "w") as f:
               f.writelines(content)
          return

     def parse_argument(self ):
          parser = argparse.ArgumentParser()
          parser.add_argument( "-g","--generateConfig", 

     help="Generate router config files.\n"

     +"This will overwrite existing files", 

     action="store_true")
          parser.add_argument("-v", "--verbose", 

     help="Prints detailed logs during network creation 
and stop", 

     action="store_true")
          flags = parser.parse_args()
          return flags

     def build(self, *args, **kwargs):
          flags = self.parse_argument()
          if(flags.verbose):
               setLogLevel( 'info' )

          # directory to keep the configurations
          config_path = "/home/riady/net101/frr-
config/%(name)s"

          # private directory that will useed by the 
routers by bind mounting
```

```
          privateDirs = [ ( '/var/log' ),
                              ( '/etc/frr',
config_path),
                              ( '/var/run' ),
                              '/var/mn' ]

          #  R1 subnet
          C1_1 = self.addHost('C1_1',
ip="172.16.1.2/24", defaultRoute="via 172.16.1.1")
          C1_2 = self.addHost('C1_2',
ip="172.16.2.2/24", defaultRoute="via 172.16.2.1")
          R1 = self.addNode("R1", cls=LinuxRouter,
ip=None, privateDirs=privateDirs, inNamespace=True)
          S1 = self.addSwitch("S1", inNamespace=True)
          R1_1 = self.addNode("R1_1", cls=LinuxRouter,
ip=None, privateDirs=privateDirs, inNamespace=True)
          R1_2 = self.addNode("R1_2", cls=LinuxRouter,
ip=None, privateDirs=privateDirs, inNamespace=True)

          # add links for subnet 1
          self.addLink(S1, R1,  intfName2="R1-eth0")
          self.addLink(S1, R1_1, intfName2="R1_1-eth0")
          self.addLink(S1, R1_2, intfName2="R1_2-eth0")

          #region
          # Do not manually set the interface name of a
switch's interface
          # mininet will not be able to automatically
add the interfaces to its bridge
          #endregion

          self.addLink(C1_1,R1_1, intfName2="R1_1-eth1")
          self.addLink(C1_2,R1_2, intfName2="R1_2-eth1")

          # R2 Subnet
          C2_1 = self.addHost('C2_1',
ip="172.17.1.2/24", defaultRoute="via 172.17.1.1")
          C2_2 = self.addHost('C2_2',
ip="172.17.2.2/24", defaultRoute="via 172.17.2.1")
          R2 = self.addNode("R2", cls=LinuxRouter,
ip=None, privateDirs=privateDirs, inNamespace=True)
          S2 = self.addSwitch("S2", inNamespace=True)

          R2_1 = self.addNode("R2_1", cls=LinuxRouter,
ip=None, privateDirs=privateDirs, inNamespace=True)
          R2_2 = self.addNode("R2_2", cls=LinuxRouter,
ip=None, privateDirs=privateDirs, inNamespace=True)
```

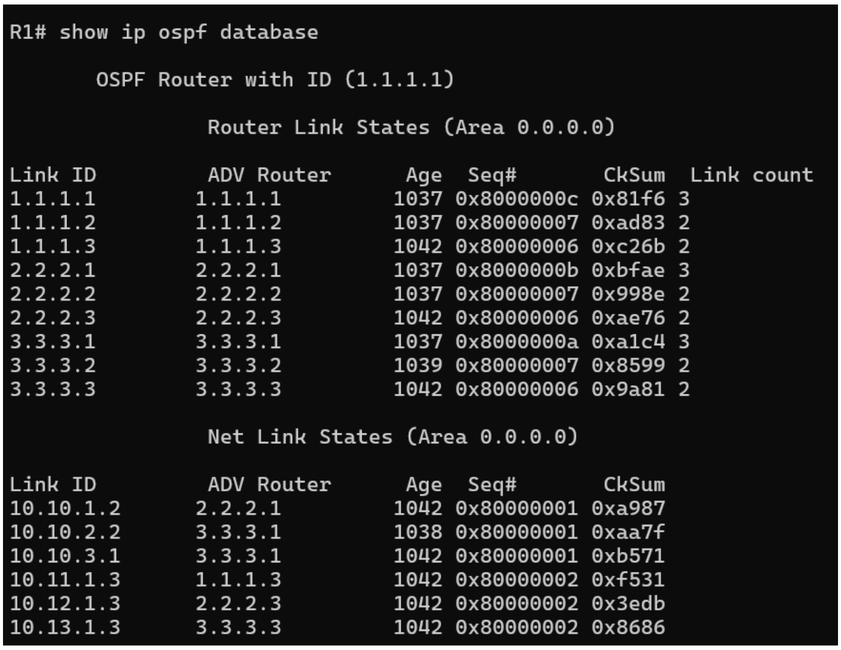
```
          # add links for subnet 2
          self.addLink(S2, R2, intfName2="R2-eth0")
          self.addLink(S2, R2_1, intfName2="R2_1-eth0")
          self.addLink(S2, R2_2, intfName2="R2_2-eth0")

          self.addLink(C2_1,R2_1, intfName2="R2_1-eth1")
          self.addLink(C2_2,R2_2, intfName2="R2_2-eth1")

          # R3 Subnet
          C3_1 = self.addHost('C3_1',
ip="172.18.1.2/24", defaultRoute="via 172.18.1.1")
          C3_2 = self.addHost('C3_2',
ip="172.18.2.2/24", defaultRoute="via 172.18.2.1")
          R3 = self.addNode("R3", cls=LinuxRouter,
ip=None, privateDirs=privateDirs, inNamespace=True)
          S3 = self.addSwitch("S3", inNamespace=True)
          R3_1 = self.addNode("R3_1", cls=LinuxRouter,
ip=None, privateDirs=privateDirs, inNamespace=True)
          R3_2 = self.addNode("R3_2", cls=LinuxRouter,
ip=None, privateDirs=privateDirs, inNamespace=True)

          # add links for subnet 3
          self.addLink(S3,R3, intfName2="R3-eth0")
          self.addLink(S3,R3_1, intfName2="R3_1-eth0")
          self.addLink(S3,R3_2, intfName2="R3_2-eth0")

          self.addLink(C3_1,R3_1, intfName2="R3_1-eth1")
          self.addLink(C3_2,R3_2, intfName2="R3_2-eth1")

          # Add links between backbone routers
          self.addLink(R1,R2, intfName1="R1-eth1",
intfName2="R2-eth1")
          self.addLink(R1,R3, intfName1="R1-eth2",
intfName2="R3-eth1")
          self.addLink(R2,R3, intfName1="R2-eth2",
intfName2="R3-eth2")

          confdir = Path(config_path % {"name": ""})
          if (not flags.generateConfig):
               if (not Path.exists(confdir)):
                    # Automatically set to generate
config files if config Path doesn't exists, such as when
first time running the program
                    print("If this is your first time
running the program, ")
```

```
                    print("consider running the program
with \"-h\" to see the options")
                    print("="*40)
                    flags.generateConfig=True

          if (flags.generateConfig):
               # Configuration files will be created for
each routers
               for n in self.nodes():
                    print(n)
                    if "cls" in self.nodeInfo(n):
                         node_info = self.nodeInfo(n)
                         if node_info["cls"].__name__ ==
"LinuxRouter":
                              self.generate_config(n,
config_path)
                              pass

          super().build(*args, **kwargs)


start = time.time()
print("This the topology for the OSPF lab")
print("="*40)
net = Mininet(topo=OSPFLab(), switch=LinuxBridge,
controller=None)
finish = time.time()
print("Finished initializing network in:", finish-start,
"seconds")

try:
     pass
     net.start()
     CLI(net)

finally:
     start = time.time()
     net.stop()
     finish = time.time()
     print("Finished stopping network in:", finish-
start, "seconds")
```

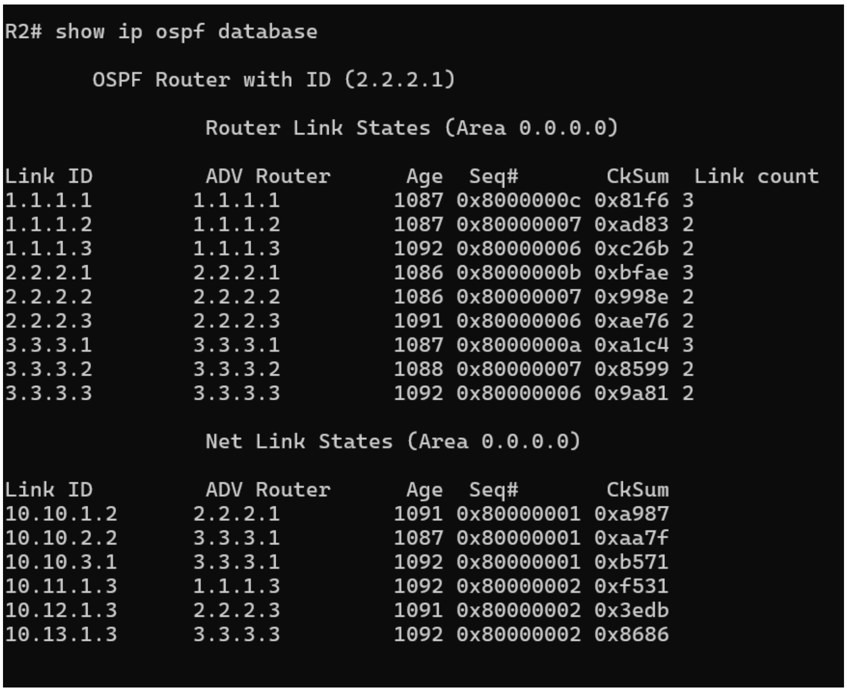
4. Pengujian dan Hasil

Langkah-langkah pengujian:

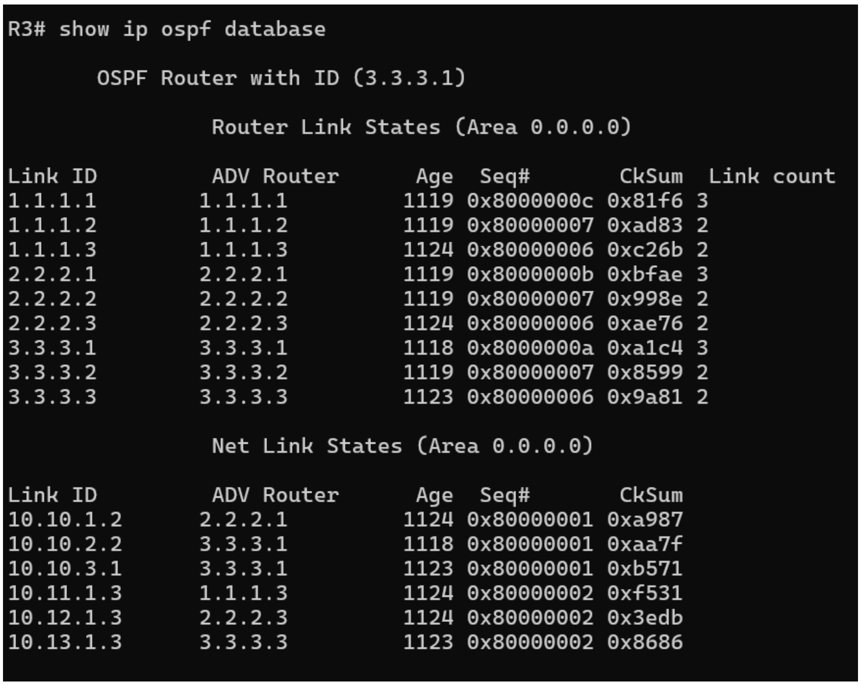
Single Area:



```
R1# show ip ospf database

       OSPF Router with ID (1.1.1.1)

                Router Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
1.1.1.1         1.1.1.1         1037 0x8000000c 0x81f6 3
1.1.1.2         1.1.1.2         1037 0x80000007 0xad83 2
1.1.1.3         1.1.1.3         1042 0x80000006 0xc26b 2
2.2.2.1         2.2.2.1         1037 0x8000000b 0xbfae 3
2.2.2.2         2.2.2.2         1037 0x80000007 0x998e 2
2.2.2.3         2.2.2.3         1042 0x80000006 0xae76 2
3.3.3.1         3.3.3.1         1037 0x8000000a 0xa1c4 3
3.3.3.2         3.3.3.2         1039 0x80000007 0x8599 2
3.3.3.3         3.3.3.3         1042 0x80000006 0x9a81 2

                Net Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum
10.10.1.2       2.2.2.1         1042 0x80000001 0xa987
10.10.2.2       3.3.3.1         1038 0x80000001 0xaa7f
10.10.3.1       3.3.3.1         1042 0x80000001 0xb571
10.11.1.3       1.1.1.3         1042 0x80000002 0xf531
10.12.1.3       2.2.2.3         1042 0x80000002 0x3edb
10.13.1.3       3.3.3.3         1042 0x80000002 0x8686
```

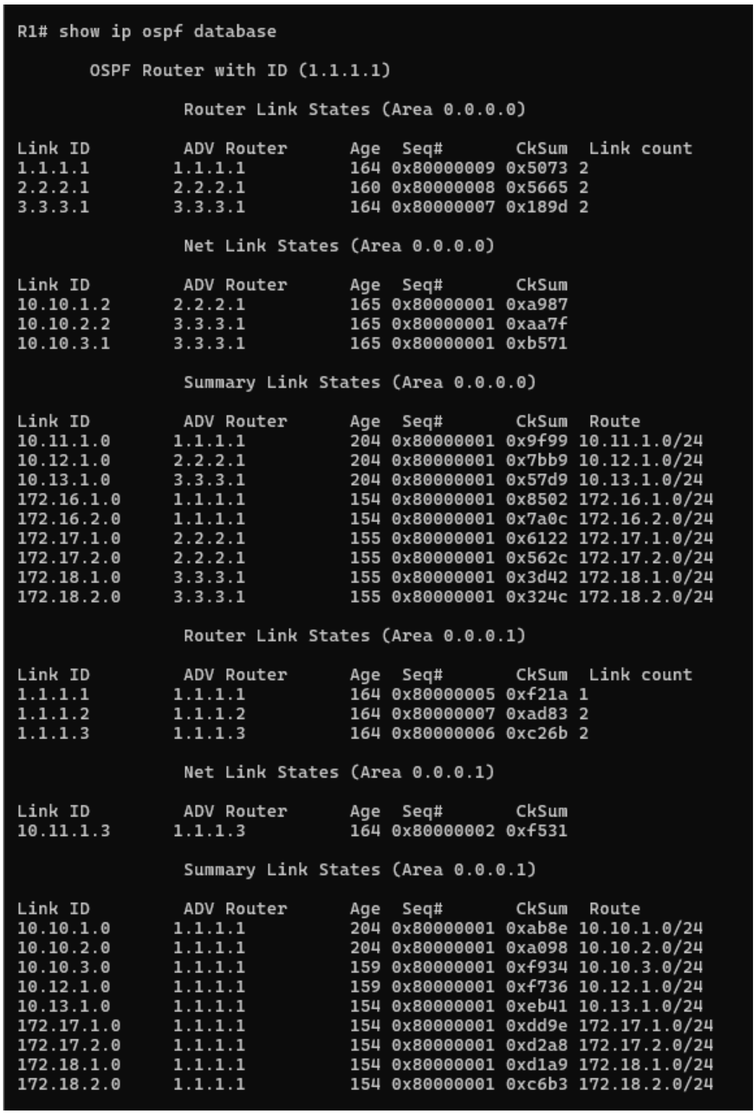


```
R2# show ip ospf database

       OSPF Router with ID (2.2.2.1)

                Router Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
1.1.1.1         1.1.1.1         1087 0x8000000c 0x81f6 3
1.1.1.2         1.1.1.2         1087 0x80000007 0xad83 2
1.1.1.3         1.1.1.3         1092 0x80000006 0xc26b 2
2.2.2.1         2.2.2.1         1086 0x8000000b 0xbfae 3
2.2.2.2         2.2.2.2         1086 0x80000007 0x998e 2
2.2.2.3         2.2.2.3         1091 0x80000006 0xae76 2
3.3.3.1         3.3.3.1         1087 0x8000000a 0xa1c4 3
3.3.3.2         3.3.3.2         1088 0x80000007 0x8599 2
3.3.3.3         3.3.3.3         1092 0x80000006 0x9a81 2

                Net Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum
10.10.1.2       2.2.2.1         1091 0x80000001 0xa987
10.10.2.2       3.3.3.1         1087 0x80000001 0xaa7f
10.10.3.1       3.3.3.1         1092 0x80000001 0xb571
10.11.1.3       1.1.1.3         1092 0x80000002 0xf531
10.12.1.3       2.2.2.3         1091 0x80000002 0x3edb
10.13.1.3       3.3.3.3         1092 0x80000002 0x8686
```

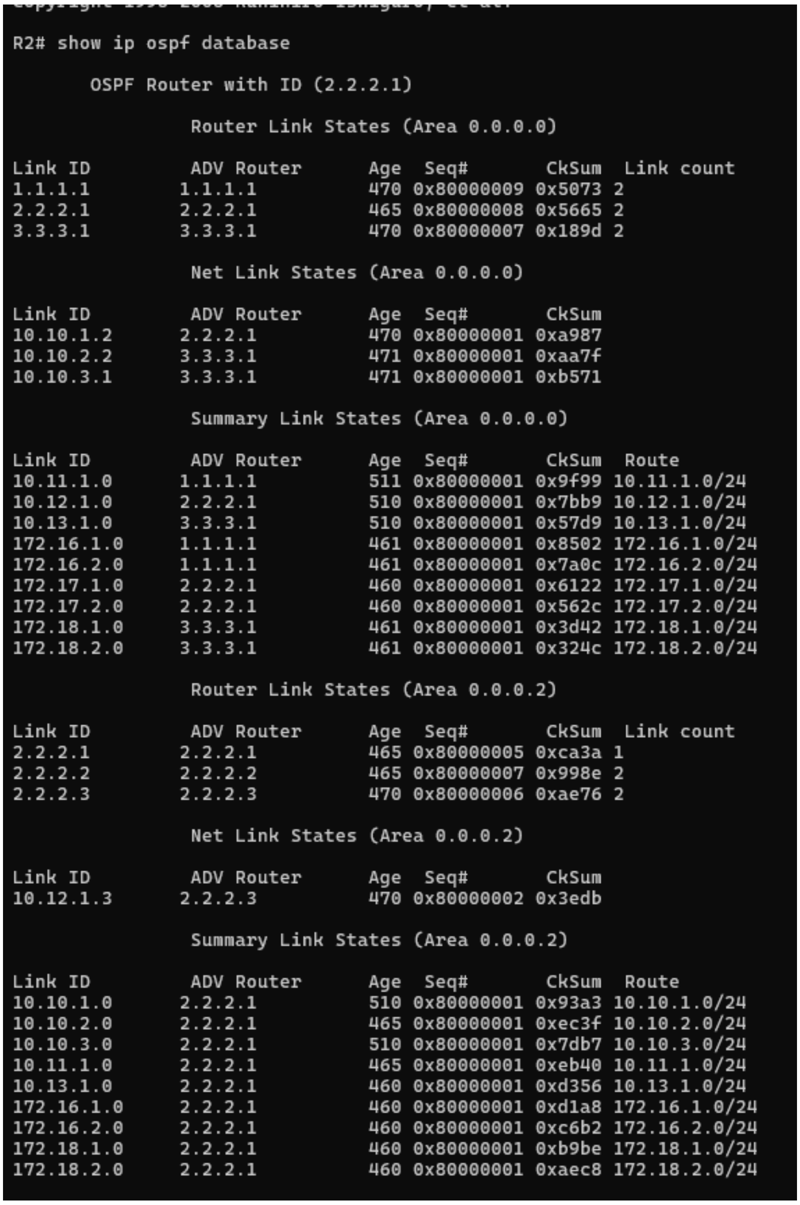


```
R3# show ip ospf database

       OSPF Router with ID (3.3.3.1)

                Router Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
1.1.1.1        1.1.1.1         1119 0x8000000c 0x81f6 3
1.1.1.2        1.1.1.2         1119 0x80000007 0xad83 2
1.1.1.3        1.1.1.3         1124 0x80000006 0xc26b 2
2.2.2.1        2.2.2.1         1119 0x8000000b 0xbfae 3
2.2.2.2        2.2.2.2         1119 0x80000007 0x998e 2
2.2.2.3        2.2.2.3         1124 0x80000006 0xae76 2
3.3.3.1        3.3.3.1         1118 0x8000000a 0xa1c4 3
3.3.3.2        3.3.3.2         1119 0x80000007 0x8599 2
3.3.3.3        3.3.3.3         1123 0x80000006 0x9a81 2

                Net Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum
10.10.1.2      2.2.2.1         1124 0x80000001 0xa987
10.10.2.2      3.3.3.1         1118 0x80000001 0xaa7f
10.10.3.1      3.3.3.1         1123 0x80000001 0xb571
10.11.1.3      1.1.1.3         1124 0x80000002 0xf531
10.12.1.3      2.2.2.3         1124 0x80000002 0x3edb
10.13.1.3      3.3.3.3         1123 0x80000002 0x8686
```

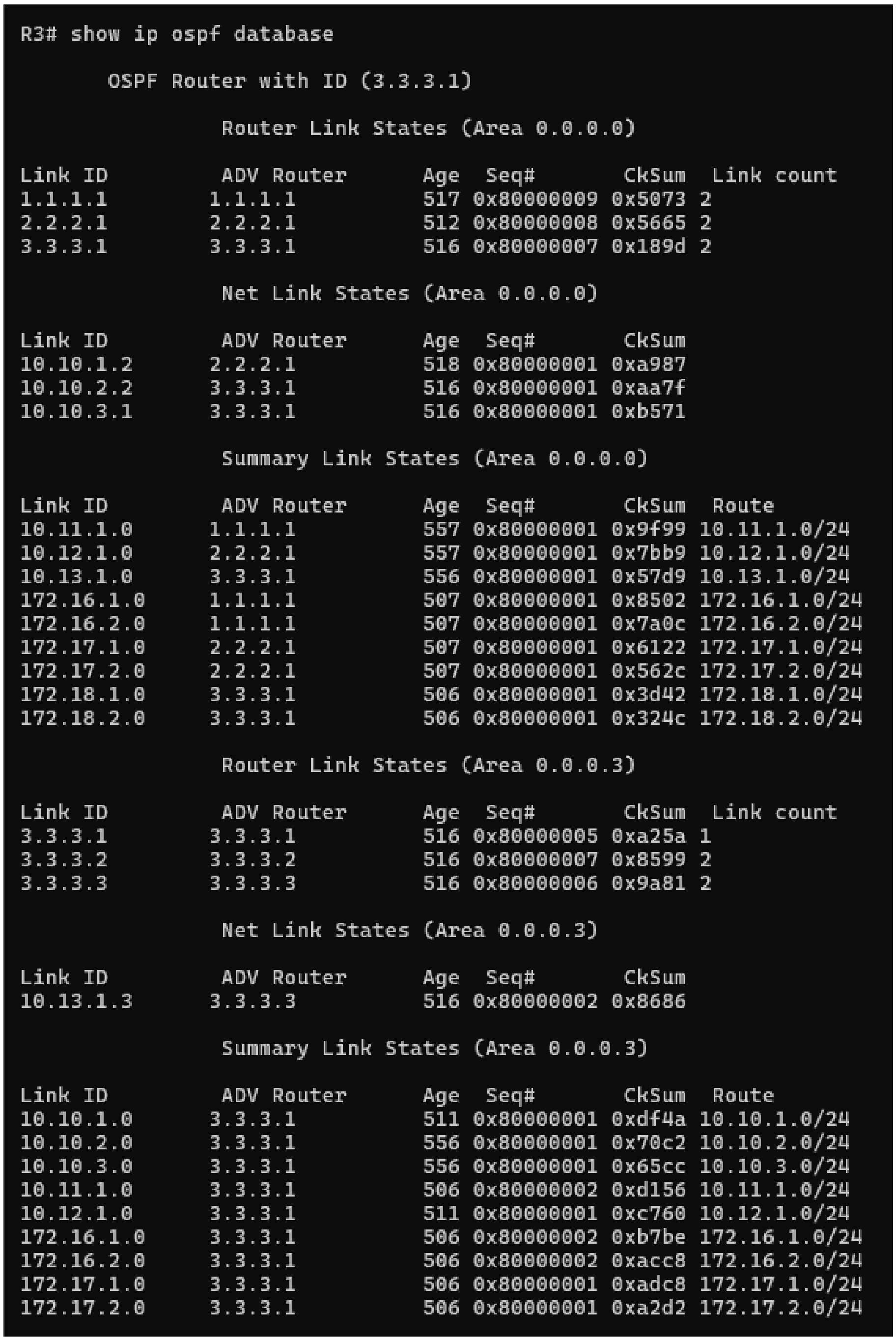
Multi Area:



```
R1# show ip ospf database

       OSPF Router with ID (1.1.1.1)

                Router Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
1.1.1.1        1.1.1.1         164 0x80000009 0x5073 2
2.2.2.1        2.2.2.1         160 0x80000008 0x5665 2
3.3.3.1        3.3.3.1         164 0x80000007 0x189d 2

                Net Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum
10.10.1.2      2.2.2.1         165 0x80000001 0xa987
10.10.2.2      3.3.3.1         165 0x80000001 0xaa7f
10.10.3.1      3.3.3.1         165 0x80000001 0xb571

                Summary Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Route
10.11.1.0      1.1.1.1         204 0x80000001 0x9f99 10.11.1.0/24
10.12.1.0      2.2.2.1         204 0x80000001 0x7bb9 10.12.1.0/24
10.13.1.0      3.3.3.1         204 0x80000001 0x57d9 10.13.1.0/24
172.16.1.0     1.1.1.1         154 0x80000001 0x8502 172.16.1.0/24
172.16.2.0     1.1.1.1         154 0x80000001 0x7a0c 172.16.2.0/24
172.17.1.0     2.2.2.1         155 0x80000001 0x6122 172.17.1.0/24
172.17.2.0     2.2.2.1         155 0x80000001 0x562c 172.17.2.0/24
172.18.1.0     3.3.3.1         155 0x80000001 0x3d42 172.18.1.0/24
172.18.2.0     3.3.3.1         155 0x80000001 0x324c 172.18.2.0/24

                Router Link States (Area 0.0.0.1)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
1.1.1.1        1.1.1.1         164 0x80000005 0xf21a 1
1.1.1.2        1.1.1.2         164 0x80000007 0xad83 2
1.1.1.3        1.1.1.3         164 0x80000006 0xc26b 2

                Net Link States (Area 0.0.0.1)

Link ID         ADV Router      Age  Seq#       CkSum
10.11.1.3      1.1.1.3         164 0x80000002 0xf531

                Summary Link States (Area 0.0.0.1)

Link ID         ADV Router      Age  Seq#       CkSum  Route
10.10.1.0      1.1.1.1         204 0x80000001 0xab8e 10.10.1.0/24
10.10.2.0      1.1.1.1         204 0x80000001 0xa098 10.10.2.0/24
10.10.3.0      1.1.1.1         159 0x80000001 0xf934 10.10.3.0/24
10.12.1.0      1.1.1.1         159 0x80000001 0xf736 10.12.1.0/24
10.13.1.0      1.1.1.1         154 0x80000001 0xeb41 10.13.1.0/24
172.17.1.0     1.1.1.1         154 0x80000001 0xdd9e 172.17.1.0/24
172.17.2.0     1.1.1.1         154 0x80000001 0xd2a8 172.17.2.0/24
172.18.1.0     1.1.1.1         154 0x80000001 0xd1a9 172.18.1.0/24
172.18.2.0     1.1.1.1         154 0x80000001 0xc6b3 172.18.2.0/24
```



```
Copyright 1996-2005 Kunihiro Ishiguro, et al.

R2# show ip ospf database

       OSPF Router with ID (2.2.2.1)

                Router Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
1.1.1.1        1.1.1.1          470 0x80000009 0x5073 2
2.2.2.1        2.2.2.1          465 0x80000008 0x5665 2
3.3.3.1        3.3.3.1          470 0x80000007 0x189d 2

                Net Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum
10.10.1.2      2.2.2.1          470 0x80000001 0xa987
10.10.2.2      3.3.3.1          471 0x80000001 0xaa7f
10.10.3.1      3.3.3.1          471 0x80000001 0xb571

                Summary Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Route
10.11.1.0      1.1.1.1          511 0x80000001 0x9f99 10.11.1.0/24
10.12.1.0      2.2.2.1          510 0x80000001 0x7bb9 10.12.1.0/24
10.13.1.0      3.3.3.1          510 0x80000001 0x57d9 10.13.1.0/24
172.16.1.0     1.1.1.1          461 0x80000001 0x8502 172.16.1.0/24
172.16.2.0     1.1.1.1          461 0x80000001 0x7a0c 172.16.2.0/24
172.17.1.0     2.2.2.1          460 0x80000001 0x6122 172.17.1.0/24
172.17.2.0     2.2.2.1          460 0x80000001 0x562c 172.17.2.0/24
172.18.1.0     3.3.3.1          461 0x80000001 0x3d42 172.18.1.0/24
172.18.2.0     3.3.3.1          461 0x80000001 0x324c 172.18.2.0/24

                Router Link States (Area 0.0.0.2)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
2.2.2.1        2.2.2.1          465 0x80000005 0xca3a 1
2.2.2.2        2.2.2.2          465 0x80000007 0x998e 2
2.2.2.3        2.2.2.3          470 0x80000006 0xae76 2

                Net Link States (Area 0.0.0.2)

Link ID         ADV Router      Age  Seq#       CkSum
10.12.1.3      2.2.2.3          470 0x80000002 0x3edb

                Summary Link States (Area 0.0.0.2)

Link ID         ADV Router      Age  Seq#       CkSum  Route
10.10.1.0      2.2.2.1          510 0x80000001 0x93a3 10.10.1.0/24
10.10.2.0      2.2.2.1          465 0x80000001 0xec3f 10.10.2.0/24
10.10.3.0      2.2.2.1          510 0x80000001 0x7db7 10.10.3.0/24
10.11.1.0      2.2.2.1          465 0x80000001 0xeb40 10.11.1.0/24
10.13.1.0      2.2.2.1          460 0x80000001 0xd356 10.13.1.0/24
172.16.1.0     2.2.2.1          460 0x80000001 0xd1a8 172.16.1.0/24
172.16.2.0     2.2.2.1          460 0x80000001 0xc6b2 172.16.2.0/24
172.18.1.0     2.2.2.1          460 0x80000001 0xb9be 172.18.1.0/24
172.18.2.0     2.2.2.1          460 0x80000001 0xaec8 172.18.2.0/24
```



```
R3# show ip ospf database

       OSPF Router with ID (3.3.3.1)

                Router Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
1.1.1.1        1.1.1.1         517 0x80000009 0x5073 2
2.2.2.1        2.2.2.1         512 0x80000008 0x5665 2
3.3.3.1        3.3.3.1         516 0x80000007 0x189d 2

                Net Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum
10.10.1.2      2.2.2.1         518 0x80000001 0xa987
10.10.2.2      3.3.3.1         516 0x80000001 0xaa7f
10.10.3.1      3.3.3.1         516 0x80000001 0xb571

                Summary Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Route
10.11.1.0      1.1.1.1         557 0x80000001 0x9f99 10.11.1.0/24
10.12.1.0      2.2.2.1         557 0x80000001 0x7bb9 10.12.1.0/24
10.13.1.0      3.3.3.1         556 0x80000001 0x57d9 10.13.1.0/24
172.16.1.0     1.1.1.1         507 0x80000001 0x8502 172.16.1.0/24
172.16.2.0     1.1.1.1         507 0x80000001 0x7a0c 172.16.2.0/24
172.17.1.0     2.2.2.1         507 0x80000001 0x6122 172.17.1.0/24
172.17.2.0     2.2.2.1         507 0x80000001 0x562c 172.17.2.0/24
172.18.1.0     3.3.3.1         506 0x80000001 0x3d42 172.18.1.0/24
172.18.2.0     3.3.3.1         506 0x80000001 0x324c 172.18.2.0/24

                Router Link States (Area 0.0.0.3)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
3.3.3.1        3.3.3.1         516 0x80000005 0xa25a 1
3.3.3.2        3.3.3.2         516 0x80000007 0x8599 2
3.3.3.3        3.3.3.3         516 0x80000006 0x9a81 2

                Net Link States (Area 0.0.0.3)

Link ID         ADV Router      Age  Seq#       CkSum
10.13.1.3      3.3.3.3         516 0x80000002 0x8686

                Summary Link States (Area 0.0.0.3)

Link ID         ADV Router      Age  Seq#       CkSum  Route
10.10.1.0      3.3.3.1         511 0x80000001 0xdf4a 10.10.1.0/24
10.10.2.0      3.3.3.1         556 0x80000001 0x70c2 10.10.2.0/24
10.10.3.0      3.3.3.1         556 0x80000001 0x65cc 10.10.3.0/24
10.11.1.0      3.3.3.1         506 0x80000002 0xd156 10.11.1.0/24
10.12.1.0      3.3.3.1         511 0x80000001 0xc760 10.12.1.0/24
172.16.1.0     3.3.3.1         506 0x80000002 0xb7be 172.16.1.0/24
172.16.2.0     3.3.3.1         506 0x80000002 0xacc8 172.16.2.0/24
172.17.1.0     3.3.3.1         506 0x80000001 0xadc8 172.17.1.0/24
172.17.2.0     3.3.3.1         506 0x80000001 0xa2d2 172.17.2.0/24
```

Analisis:

Dalam konfigurasi OSPF single-area, seluruh jaringan berada dalam satu area, yaitu Area 0 (backbone area), yang memberikan visibilitas penuh kepada semua router terhadap jaringan secara keseluruhan. Hasil traceroute dari single-area menunjukkan lima hop dari C1_1 ke C2_1 dengan latensi antara 0,088 ms hingga 1,098 ms. Hal ini menandakan jalur yang cukup efisien, di mana setiap router memiliki pengetahuan lengkap tentang semua rute karena tidak ada pembagian area. Database OSPF yang ditampilkan dari router dalam single-area menunjukkan bahwa router memiliki informasi menyeluruh mengenai tautan dan status jaringan secara keseluruhan.

Sementara itu, dalam konfigurasi OSPF multi-area, jaringan dibagi menjadi beberapa area (seperti Area 0.0.0.0 dan Area 0.0.0.1), yang mengurangi jumlah informasi yang perlu disimpan oleh setiap router, karena mereka hanya perlu mengetahui rute antar-area dan rute ringkasan. Pada traceroute untuk multi-area, terdapat lima hop dengan latensi yang serupa dengan single-area, menunjukkan kinerja yang hampir sama, tetapi struktur OSPF menjadi lebih tersegmentasi. Di database OSPF multi-area, terlihat router summary links yang hanya memberikan informasi tentang rute antar-area, serta Router Link States yang lebih spesifik untuk setiap area.

Perbedaan utama antara single-area dan multi-area adalah bahwa dalam single-area, setiap router memiliki visibilitas lengkap terhadap semua tautan dalam jaringan, sedangkan dalam multi-area, informasi dirangkum antar area untuk mengurangi ukuran database routing dan meningkatkan skalabilitas. Router dalam multi-area memiliki ringkasan rute (summary links) untuk area lain, yang tidak terlihat di database single-area karena seluruh jaringan terletak dalam satu area.

5. Kesimpulan

Konfigurasi OSPF single-area memberikan semua router akses penuh ke informasi routing di seluruh jaringan, sehingga menghasilkan jalur komunikasi yang efisien. Namun, hal ini dapat menyebabkan peningkatan ukuran database routing pada jaringan yang lebih besar. Di sisi lain, konfigurasi multi-area membagi jaringan menjadi beberapa area, membatasi informasi routing yang dapat diakses oleh setiap router hanya pada area lokalnya dan merangkum rute antar-area. Pendekatan ini meningkatkan skalabilitas jaringan dengan mengurangi ukuran database routing dan memungkinkan manajemen jaringan yang lebih efisien, meskipun kinerja komunikasi tetap hampir sama dengan konfigurasi single-area.