OSPF

OSPF (IPv4: OSPFv2 / IPv6:OSPFv3)

- **Link-state** (network prefix, prefix length, cost) routing protocol. Alternative to RIP.

- Faster convergence. Scales to much large network implementations.

- Router builds topology table using **Dijkstra Shortest-Path First (SPF) algorithm** (cumulative cost to reach a destination) to calculate best routes.

- **Areas**: divide the routing domain into areas that help control routing update traffic.
  - **Single-Area OSPF**: all routers in one area. **Area 0**.
  - **Multiarea OSPF**:
    - All areas connected to backbone area (Area 0)
    - Area Border Routers (ABRs): routers interconnecting the areas.
    - Smaller routing tables. Reduced link-state update overhead. Reduced frequency of SPF calculations.

Types of OSPF Packets

5 types of packets (LSP: Link State Packets) to discover neighbors and exchange routing info:

1️⃣ **Hello packet**: Discovers neighbors and builds adjacencies between them
2️⃣ **Database description (DBD)**: Checks for database synchronization between routers
3️⃣ **Link-state request (LSR)**: Requests specific link-state records from router to router
4️⃣ **Link-state update (LSU)**: Sends specifically requested link-state records
5️⃣ **Link-state acknowledgment packet (LSack)**: Acknowledges the other packet types

Components of OSPF - Databases

3 OSPF databases:

1 **Adjacency Database** (Neighbor Table) `show ip ospf database`
List of all neighbor routers. Unique for each router.

2 **Link-state Database (LSDB)** (Topology Table) `show ip ospf database`
Lists info about all other routers. All routers within an area have same LSDB.

3 **Forwarding Database** (Routing Table)
List of routers generated when algorithm is run. Unique router's routing table.

Components of OSPF - SPF algorithm

1 Router builds topology table using Dijkstra shortest-path first (SPF) algorithm (cumulative cost to reach a destination) to calculate best routes.
2 OSPF places the best routes into the Forwarding Database.

Link-State Operation

Link-state routing steps that are completed by a router:

1 Establish Neighbor Adjacencies
2 Exchange Link-State Advertisements
3 Build the Link State Database
4 Execute the SPF Algorithm
5 Choose the Best Route

OSPF Terminology

- **Adjacency:** is the relationship that must be formed between two OSPF routers that allows them to exchange routing table updates. Adjacencies can only be formed by routers that are in the same area.

- **Router ID:** is a unique identifier used by OSPF routers to communicate with each other. It can be set by using the router-idcommand, but if it's not set the highest IP address on the loopback interface will become your router-id, if the loopback is not set then it's the highest IP on a physical interface.

OSPF Terminology

- **Designated Router:** is determined by the priority set on the router (highest number wins). If the priority matches then the Router-id is the tie breaker (highest Router-id wins). The designated router (DR) forms adjacencies to all routers so it can send and receive all routing table updates.

- **Backup Designated Router:** is there as a backup for the DR. It's the router with the next highest priority or the next highest Router-ID. It receives all routing table updates but does NOT send them out.

- **Hello Protocol:** used to discover other OSPF routers and maintains the adjacency between them.

- **Passive Interface:** to prevent the transmission of routing messages through a router interface, but still allow that network to be advertised to other routers.

OSPF Terminology

- **Link State Advertisement:** is an OSPF data packet that sends out link-state and routing information. Only routers that form an adjacency can exchange link state advertisements (LSA)

- **Neighborship database:** list of all OSPF routers. This is a unique feature of OSPF.

- **Topological database:** list of all OSPF paths (routes).

- **Multi-access (Broadcast) / Non-broadcast multi-access (NBMA)**: are types of networks that require a DR and a BDR. This is determined by the type of WAN link you're connected to.

- **Point-to-Point / Point-to-multipoint**: Do NOT require a DR or BDR. This is determined by the type of WAN link you're connected to.

OSPF Verification Commands

`show ip ospf` ➡️ Display OSPF Configuration

`show ip ospf database` ➡️ Display the Link ID of all routers and which neighbor is advertising that router to you

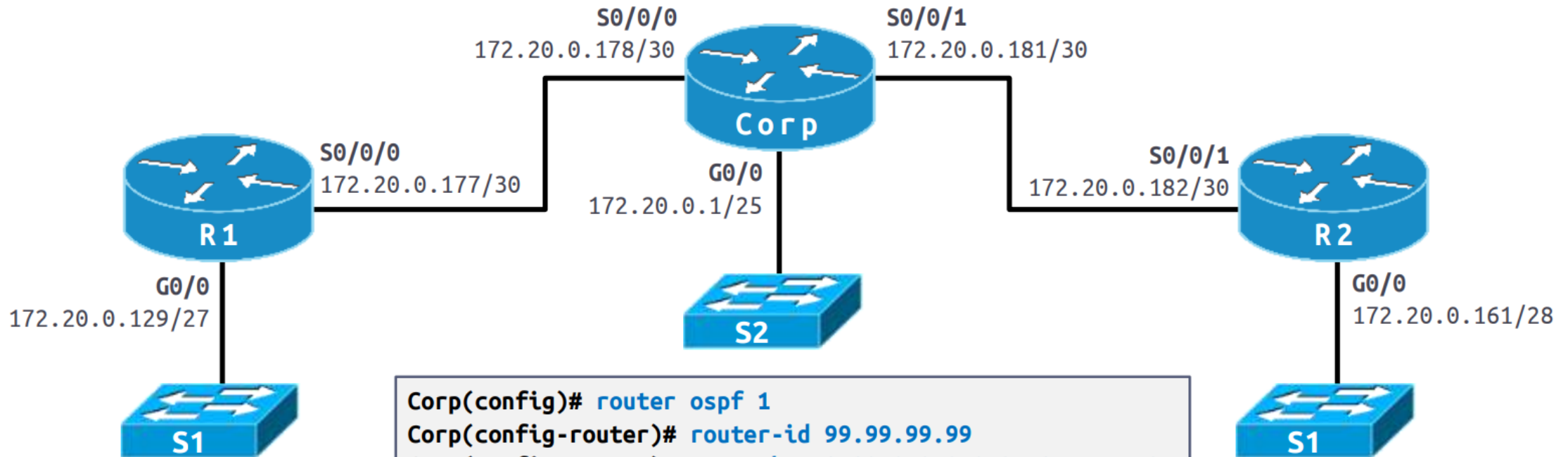`show ip ospf neighbor` ➡️ Display your OSPF neighbors

`show ip ospf interface` ➡️ Display each interfaces IP, Area, Process ID, Router ID, Network type, Cost, Priority, DR/BDR election, and timers. `show ip protocols` ➡️ Display OSPF configurations you have set, Process ID, Router ID, Network advertisements.

Propagate a Default Static Route in OSPFv2

```
R2(config)# router ospf 10
R2(config-router)# default-information originate
```

OSPFv2 IPv4 Single Area

Option 1: OSPFv2 IPv4 (Single Area) by Advertising

S0/0/0
172.20.0.178/30

S0/0/1
172.20.0.181/30

Corp

S0/0/0
172.20.0.177/30

G0/0
172.20.0.1/25

S0/0/1
172.20.0.182/30

R1

R2

G0/0
172.20.0.129/27

G0/0
172.20.0.161/28
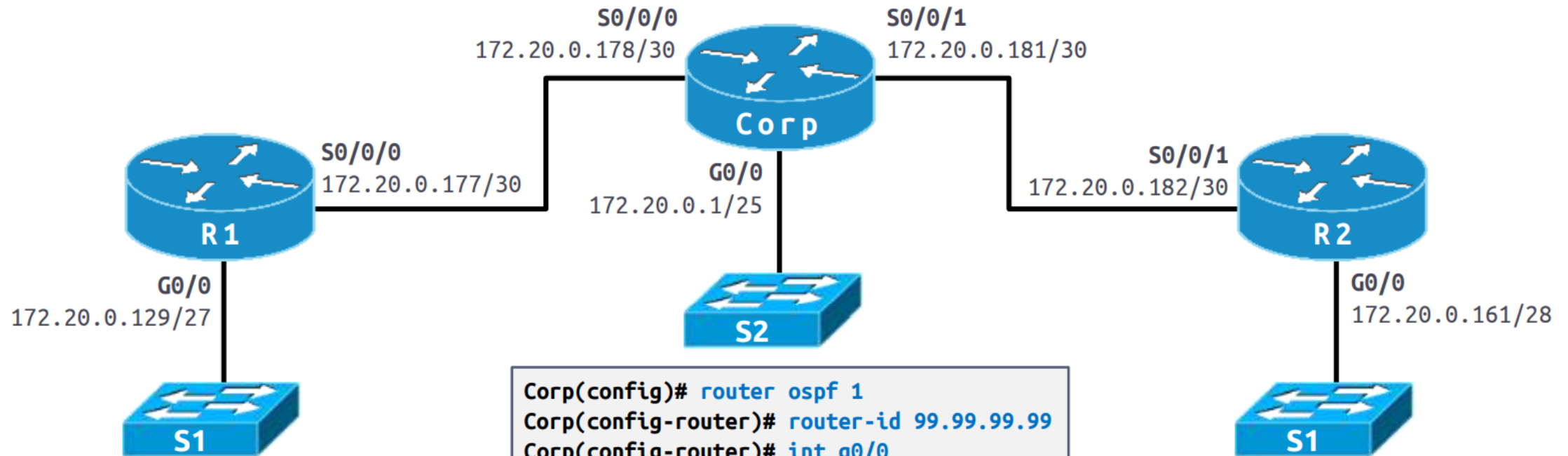
S1

S2

S1

```
Corp(config)# router ospf 1
Corp(config-router)# router-id 99.99.99.99
Corp(config-router)# network 172.20.0.0 0.0.0.127 area 0
Corp(config-router)# network 172.20.0.176 0.0.0.3 area 0
Corp(config-router)# network 172.20.0.180 0.0.0.3 area 0
```

```
R1(config)# router ospf 1
R1(config-router)# router-id 1.1.1.1
R1(config-router)# network 172.20.0.128 0.0.0.31 area 0
R1(config-router)# network 172.20.0.176 0.0.0.3 area 0
```

```
R2(config)# router ospf 1
R2(config-router)# router-id 2.2.2.2
R2(config-router)# network 172.20.0.160 0.0.0.15 area 0
R2(config-router)# network 172.20.0.180 0.0.0.3 area 0
```

Option 2: OSPFv2 IPv4 (Single Area) by Interface

S0/0/0
172.20.0.178/30

S0/0/1
172.20.0.181/30

Corp

S0/0/0
172.20.0.177/30

G0/0
172.20.0.1/25

S0/0/1
172.20.0.182/30

R1

R2

G0/0
172.20.0.129/27

G0/0
172.20.0.161/28

S1

S2

S1

```
Corp(config)# router ospf 1
Corp(config-router)# router-id 99.99.99.99
Corp(config-router)# int g0/0
Corp(config-if)# ip ospf 1 area 0
Corp(config-if)# int s0/0/0
Corp(config-if)# ip ospf 1 area 0
Corp(config-if)# int s0/0/1
Corp(config-if)# ip ospf 1 area 0
```
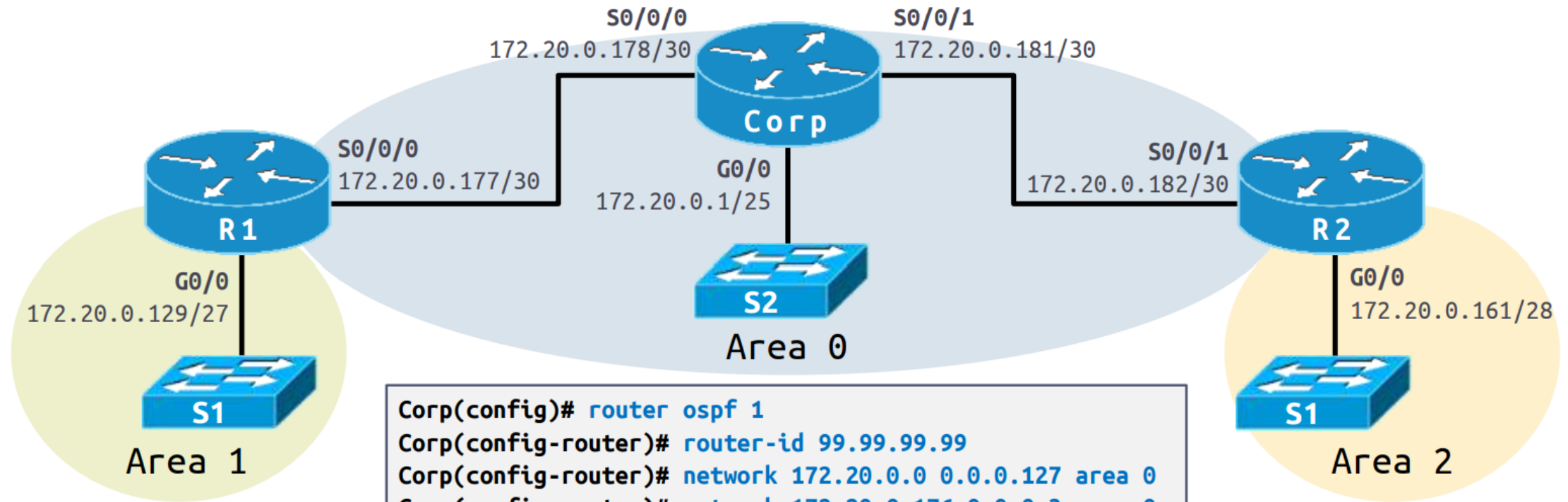
```
R1(config)# router ospf 1
R1(config-router)# router-id 1.1.1.1
R1(config-router)# int g0/0
R1(config-if)# ip ospf 1 area 0
R1(config-if)# int s0/0/0
R1(config-if)# ip ospf 1 area 0
```

```
R2(config)# router ospf 1
R2(config-router)# router-id 2.2.2.2
R2(config-router)# int g0/0
R2(config-if)# ip ospf 1 area 0
R2(config-if)# int s0/0/1
R2(config-if)# ip ospf 1 area 0
```

OSPFv2 IPv4 Multi-Area

OSPFv2 IPv4 (Multi-Area)

S0/0/0
172.20.0.178/30

S0/0/1
172.20.0.181/30

Corp

S0/0/0
172.20.0.177/30

G0/0
172.20.0.1/25

S0/0/1
172.20.0.182/30

R 1

R 2

G0/0
172.20.0.129/27

G0/0
172.20.0.161/28
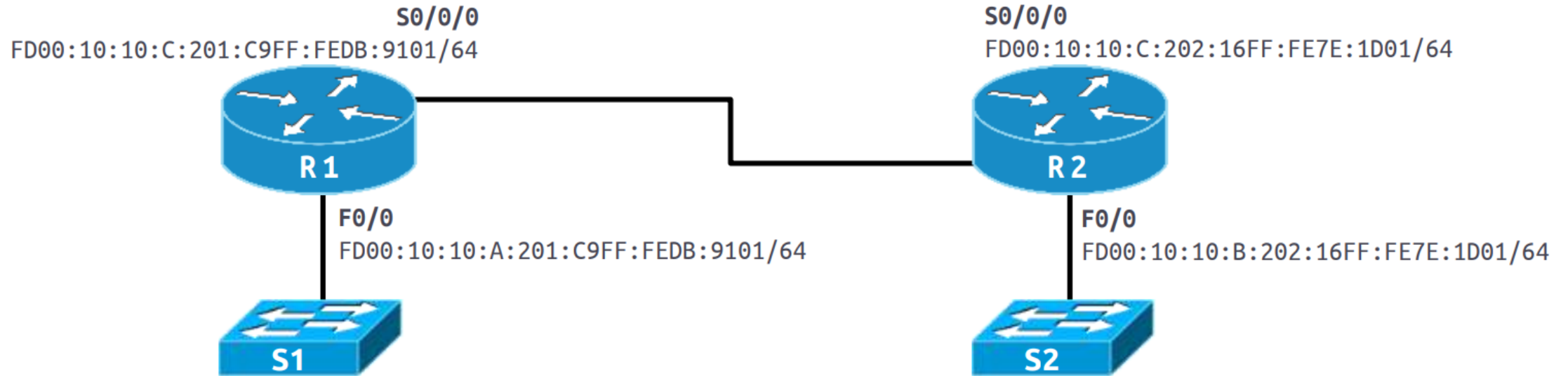
S2

Area 0

S1

S1

Area 1

Area 2

```
Corp(config)# router ospf 1
Corp(config-router)# router-id 99.99.99.99
Corp(config-router)# network 172.20.0.0 0.0.0.127 area 0
Corp(config-router)# network 172.20.0.176 0.0.0.3 area 0
Corp(config-router)# network 172.20.0.180 0.0.0.3 area 0
```

```
R1(config)# router ospf 1
R1(config-router)# router-id 1.1.1.1
R1(config-router)# network 172.20.0.128 0.0.0.31 area 1
R1(config-router)# network 172.20.0.176 0.0.0.3 area 0
```

```
R2(config)# router ospf 1
R2(config-router)# router-id 2.2.2.2
R2(config-router)# network 172.20.0.160 0.0.0.15 area 2
R2(config-router)# network 172.20.0.180 0.0.0.3 area 0
```

OSPFv3 IPv6

OSPFv3 IPv6

S0/0/0
FD00:10:10:C:201:C9FF:FEDB:9101/64

S0/0/0
FD00:10:10:C:202:16FF:FE7E:1D01/64

R1

R2

F0/0
FD00:10:10:A:201:C9FF:FEDB:9101/64

F0/0
FD00:10:10:B:202:16FF:FE7E:1D01/64

S1

S2

R1 CLI
```
R1(config)# ipv6 unicast-routing
R1(config)# ipv6 router ospf 1
R1(config-rtr)# router-id 1.1.1.1
R1(config-rtr)# passive-interface f0/0
R1(config)# int f0/0
R1(config-if)# ipv6 ospf 1 area 0
R1(config)# int s0/0/0
R1(config-if)# ipv6 ospf 1 area 0
```

R2 CLI
```
R2(config)# ipv6 unicast-routing
R2(config)# ipv6 router ospf 1
R2(config-rtr)# router-id 2.2.2.2
R2(config-rtr)# passive-interface f0/0
R2(config)# int f0/0
R2(config-if)# ipv6 ospf 1 area 0
R2(config)# int s0/0/0
R2(config-if)# ipv6 ospf 1 area 0
```