ENARSI

November 20, 2020

1 Chapter 1: L3 Technologies

1.1 Admin Distance

- Belivability of a routing advertisement

AD	Routing Source
0	Connected
1	Static (default)
20	eBGP
90	Internal EIGRP (D)
110	OSPF
115	IS-IS
120	RIP
170	External EIGRP (D EX)
200	iBGP
255	Static default over DHCP

- Verify by ip Routing table `sh ip route` each prefix has `[AD/Metric]` values attached
- To alter AD of static route using **Floating Static Route**: `ip route DEST_IP NH_IP DIST`

1.2 Redistribution basics

1.2.1 Needs of redistribution

- Single AS runnning multiple IGP
- IGP communicated with BGP and vise-versa

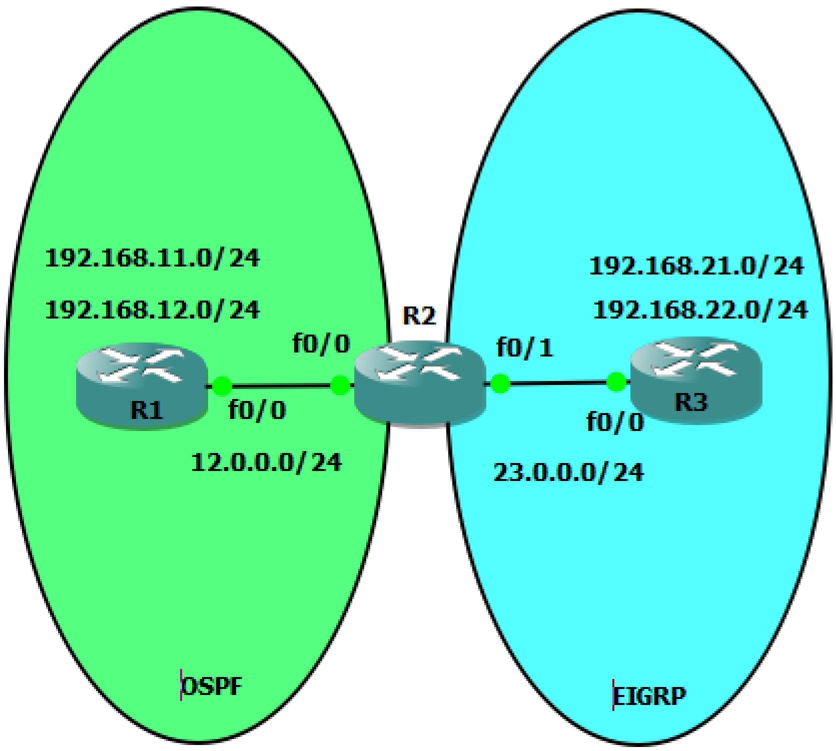
1.2.2 Challenges

- Each routing protocol has its own metric. While redistributed each RP uses a default **Seed Metric**.

Routing Protocol	Seed Metric
RIP	∞

Routing Protocol	Seed Metric
EIGRP	∞
OSPF	20 (1 for BGP)
BGP	Uses IGP Metric

1.3 Mutual Route-Redistribution



Router R2 runs both OSPF and EIGRP with appropreate settings. However, R1 is not aware about R3 routes and vice-versa. In **Mutual Route-Redistribution** R2 will mutually redistribute routes learnt from R1 and R3 among each other.

```
R2#sh ip route ospf
     192.168.12.0/32 is subnetted, 1 subnets
O       192.168.12.1 [110/2] via 12.0.0.1, 00:01:41, FastEthernet0/0
     192.168.11.0/32 is subnetted, 1 subnets
O       192.168.11.1 [110/2] via 12.0.0.1, 00:01:41, FastEthernet0/0

R2#sh ip route eigrp
D    192.168.21.0/24 [90/156160] via 23.0.0.3, 00:00:34, FastEthernet0/1
D    192.168.22.0/24 [90/156160] via 23.0.0.3, 00:00:34, FastEthernet0/1
```

Syntax

```
conf t
```

```
    router TARGET_RP
        redistribute SOURCE_RP
end
```

- Redistribution EIGRP AS 1 into OSPF in classless mode with custom metric 40

```
!r2
router ospf 1
    redistribute eigrp 1 subnets metric 40

!r1
sh ip route ospf
     23.0.0.0/24 is subnetted, 1 subnets
O E2    23.0.0.0 [110/40] via 12.0.0.2, 00:00:28, FastEthernet0/0
O E2 192.168.21.0/24 [110/40] via 12.0.0.2, 00:00:28, FastEthernet0/0
O E2 192.168.22.0/24 [110/40] via 12.0.0.2, 00:00:28, FastEthernet0/0
```

- Redistribute OSPF into EIGRP AS 1 with appropreate metric. Check the metric=170 for DEX routes.

```
!r2
router eigrp 1
    redistribute ospf 1 metric 1000000 10 255 1 1500

!r3
R3#sh ip route eigrp
     192.168.12.0/32 is subnetted, 1 subnets
D EX    192.168.12.1 [170/30720] via 23.0.0.2, 00:00:08, FastEthernet0/0
     192.168.11.0/32 is subnetted, 1 subnets
D EX    192.168.11.1 [170/30720] via 23.0.0.2, 00:00:08, FastEthernet0/0
     12.0.0.0/24 is subnetted, 1 subnets
D EX    12.0.0.0 [170/30720] via 23.0.0.2, 00:00:08, FastEthernet0/0
```

1.4 Altering seed metric

- **Option 1: gloabally for a target routing proto** where all redistributed routes will have identical metric, regardless of their source. Make sure put the configuration ont he distributing router.

```
router ospf|eigrp ASN
    default-metric [ARG1, ARG2,...]
```

- **Option 2: in `redistribute` command** where the metric is specified int the `redistribute` command with `metric` followed by RP-specific parameters.
- **Option 3: using `route-map`** where metric alteration is dictated by a routing policy.

```
conf t
    route-map MAP_REDIST !create a RMAP for any match
        set metric 1000000 1 255 1 1500
    exit
```

```
    router eigrp 1  !apply the RMAP to the EIGRP redist
        redist ospf 1 route-map MAP_REDIST
end
```

- E1 and E2 route in OSPF: OSPF redistributes with metric-type 2 (E2) by default that does not change while hopping over IGP domain, to make it use E1. using `redist SRC_RP subnets metric VAL metric-type 1|2` command.

2 Route Filtering

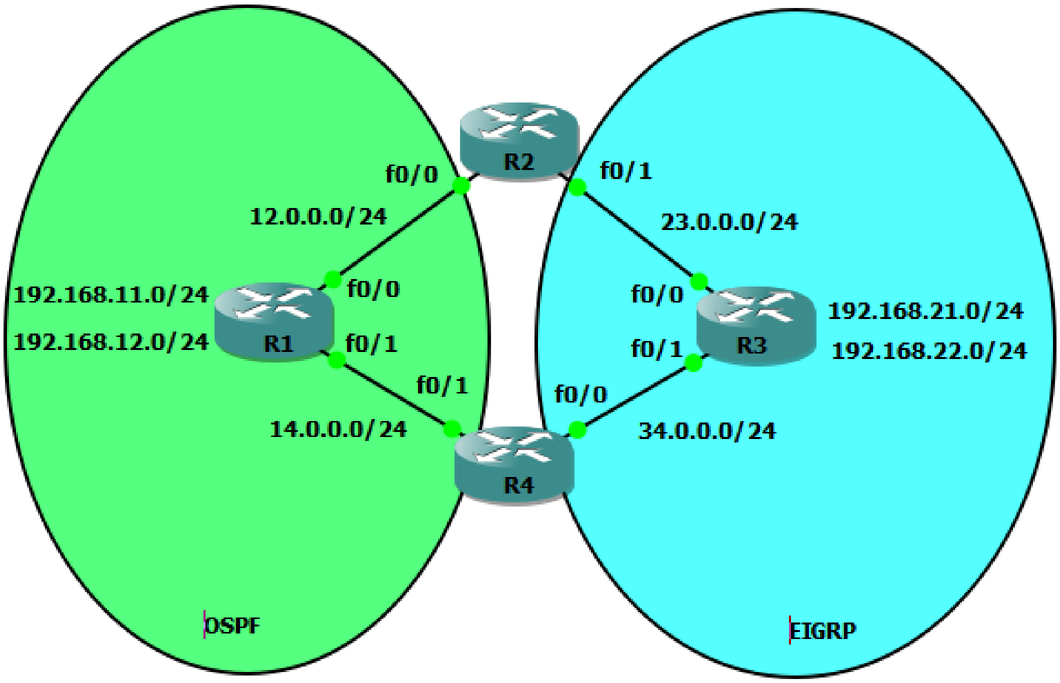
Filter or choose which routes are permitted/denied to get redistributed.

TASK: Allow only 192.168.11.0/24 and 192.168.21.0/24 routes to be redistributed.

```
conf t
    !create an ACL of permitted routes
    ip access-list stand ROUTE_FILTER_LIST
        10 permit 192.168.11.0 0.0.0.255
        20 permit 192.168.21.0 0.0.0.255
    exit

    !create a route-map from the ACL
    route-map ROUTE_FILTER_MAP permit 10
        match ip address ROUTE_FILTER_LIST   ! if matches the ACL prefixes
            set metric 1000000 10 255 1 1500 ! set the metric also
    exit
    route-map ROUTE_FILTER_MAP deny 20       ! deny for others
    exit

    ! apply route filtering on the RPs
    router ospf 1
        redist eigrp 1 subnets route-map ROUTE_FILTER_MAP
    router eigrp 1
        redist ospf 1 route-map ROUTE_FILTER_MAP
end
```

2.1 Routing Loop

There are two problems that come with redistribution * **Routing Loop:** Routing updates flooded in loop within the routing domain * To prevent any avdertisement to come back the source routing domain (AS) **Route-Tagging** is used. Each redistributed routes is tagged by the boundary router and it also prevents routes with same tag to enter (**NO-RETURN**) * For example, while redistributing routes from OSPF to EIGRP domain, R2 tags them with **TAG=10** and R4 is going to deny any tag=10 from entering. * **Suboptimal Routing:** Routers chooses non-best path for preferring a slow link with a higher AD routing protocol over a fast link with a lower AD RP



In the figure, R2 recived 14.0.0.0/24 via R1 over internal OSPF (AD=110) and via R3 over external EIGRP (AD=170). therefore, R2 chooses 12.0.0.1 as a next hop than 23.0.0.3. You can verify the entry of 14.0.0.0/24 in both OSPF LSDB and EIGRP topology table. However, the routing table will always choose the best route.

```
!R2
sh ip eigrp topology | sec 14.0.0.0
P 14.0.0.0/24, 1 successors, FD is 5120
        via Redistributed (5120/0)

sh ip os database | in 14.0.0.1
14.0.0.1        192.168.12.1    1468        0x80000003 0x0025F0
```

Now if you shut the F0/0 interface of R2, it should reconverge by choosing a path via R3.

```
!r2
!after shutting down f0/0
trace 14.0.0.1

Type escape sequence to abort.
Tracing the route to 14.0.0.1

  1 23.0.0.3 60 msec 32 msec 28 msec
  2 34.0.0.4 64 msec 60 msec 64 msec
  3 14.0.0.1 80 msec 72 msec 108 msec

!
!after no-shutting f0/0
```

```
trace 14.0.0.1

Type escape sequence to abort.
Tracing the route to 14.0.0.1

  1 12.0.0.1 32 msec 28 msec 28 msec
```

2.1.1 Applying Route-Tag

Apply route-tag there must te two route-map. One will tag the routes (TAG_20) another will deny the routes with tag=20 and allow others (DENY_TAG_20). Finally apply the route-maps to the appropreate direction, i.e. tagging 20 while redistributing into EIGRP from OSPF and denying tag 20 while redistribution back to OSPF from EIGRP.

```
!R2
conf t
    route-map TAG_20 permit  !set tag=20
        set tag 20
    exit

    route-map DENY_TAG_20 deny 10  ! deny tag=20
        match tag 20
    route-map DENY_TAG_20 permit 20 ! permit any
    exit

    !applying route-tags into redistribution
    router eigrp 1
        redist ospf 1 route-map TAG_20 metric 1000000 10 255 1 1500
    router ospf 1
        redist eigrp 1 subnets route-map DENY_TAG_20
end
```

to verify the tagged routes, check the EIGRP topology table and check only redistributed routes from OSPF is tagged with 20.

```
!r2
sh ip eigrp topology | in tag

P 12.0.0.0/24, 1 successors, FD is 5120, tag is 20
P 14.0.0.0/24, 1 successors, FD is 5120, tag is 20
P 192.168.11.1/32, 1 successors, FD is 5120, tag is 20
P 192.168.12.1/32, 1 successors, FD is 5120, tag is 20
```

do the same configuration in R4.

2.2 Redistribution on IPv6

- IPv6 redistribution does not include any Connected routes.

- Explicit command **`include-connected`** is needed to include connected routes (Logic: Only foreign routes are feasible to be redistributed).
- The subnet option is implicit for OSPFv3 as IPv6 has no concept of subnets.

```
conf t
    ipv6 router eigrp 1
        redistribute ospf 1 metric 1000000 10 255 1 1500 include-connected
    ipv6 router ospf 1
        redistribute eigrp 1 include-connected
end
```

2.3 Troubleshooting route-redistribution

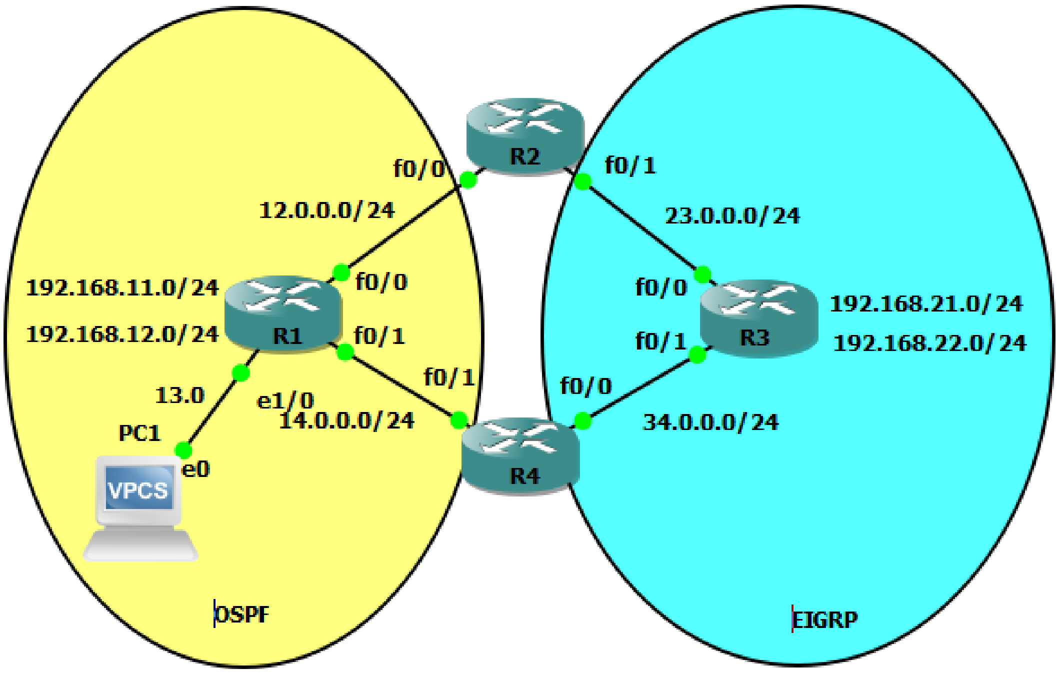
- Does the border router run both the protocol with proper AS/PID
- Check the routing table if both routes are visible in R2
- Check if the redistribution is appltied (check running confing router section)
- Check if the default metric set for non-ospf protocols
- if default-metric is not set then put metric command in reditribute statement or in a route-map.

2.4 Policy Based Routing (PBR)

- Overrides the router's default forwarding decission using custom logic
- Uses ACL to match a prefixes and a Route-Map to match and take action defined in the ACL
- Common operations

Operation	Attributes
Match	IP Address
	Range of packet Length
Set	Next hop IP Address
	Default Next-hop IP
	Egress Interface
	Default interface

- PBR is not applied by default on any traffic that is originated from the subjected router, to make it happen a varient of PBR (Local-PBR) is to be used.



2.4.1 Task:

- Configure R1 in such a way that any traffic going to 192.168.21.0 and 192.168.22.0 network must take R2 and R4 as a next hop respectively.
- **Steps**:
  - Create a ACLs to match the routes
  - Create a single route-map with two match statements, each matching individual ACL and set their appropreate actions.
  - Apply the policy to the ingress interface with `ip policy route-map MAP_NAME` command
- **Confiuration**:

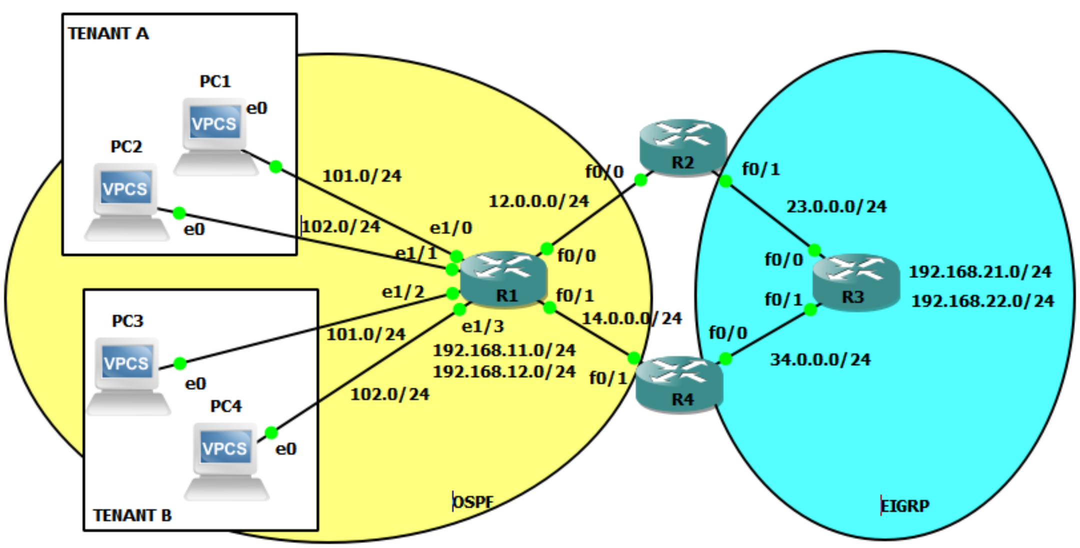
```
!r1
    conf t
        ! ACL for matching
        ip access-list ext PBR_21
            10 permit ip any 192.168.21.0 0.0.0.255
        ip access-list ext PBR_22
            10 permit ip any 192.168.22.0 0.0.0.255

        ! route-map for Policy definition
        route-map EXIT-POLICY permit 10
            match ip address PBR_21
                set ip next-hop 12.0.0.2
        exit
        route-map EXIT-POLICY permit 20
```

```
        match ip address PBR_22
            set ip next-hop 14.0.0.4
    exit

    ! apply policy to ingress interface
    int e1/0
        ip policy route-map EXIT-POLICY
```

- **Verification**:

```
PC1> trace 192.168.21.1
trace to 192.168.21.1, 8 hops max, press Ctrl+C to stop
 1   192.168.13.1   17.229 ms  14.715 ms  14.755 ms
 2   12.0.0.2   47.834 ms  45.625 ms  47.074 ms
 3   *23.0.0.3   77.558 ms (ICMP type:3, code:3, Destination port unreachable)

PC1> trace 192.168.22.1
trace to 192.168.22.1, 8 hops max, press Ctrl+C to stop
 1   192.168.13.1   16.773 ms  15.319 ms  14.704 ms
 2   14.0.0.4   46.325 ms  45.992 ms  46.096 ms
 3   *34.0.0.3   75.496 ms (ICMP type:3, code:3, Destination port unreachable)
```

2.5 Virtual Routing & Forwarding (VFR)



* Creates virtual routing instances on top of a physical router * ISP needs to maintain individual routing instances for individual customer to seperate overlapping routes, this leverages multi-tanancy. * The virtual routing instances creates logical Routing table along with vRIB and vFIB along with the global routing table with RIB and FIB. * **VRF Lite** is a special varient that does not include MPLS * VRFs run in isolation by default, however routes can be injected among them explicitly by **Route-Leaking** * **Config**:

```
conf t
```

```
    ! creating VRF
    vrf definition TENANT_A
        ip address-family ipv4
    vrf definition TENANT_B
        ip address-family ipv4

    ! registering interfaces into VRF
    ! notice TENANT_A and B has overlapping address
    int e1/0
        vrf forwarding TENANT_A
        ip add 192.168.101.1 255.255.255.0
        no sh
    int e1/1
        vrf forwarding TENANT_A
        ip add 192.168.102.1 255.255.255.0
        no sh
    int e1/2
        vrf forwarding TENANT_B
        ip add 192.168.101.1 255.255.255.0
        no sh
    int e1/3
        vrf forwarding TENANT_B
        ip add 192.168.102.1 255.255.255.0
        no sh
end
```

[ ]: