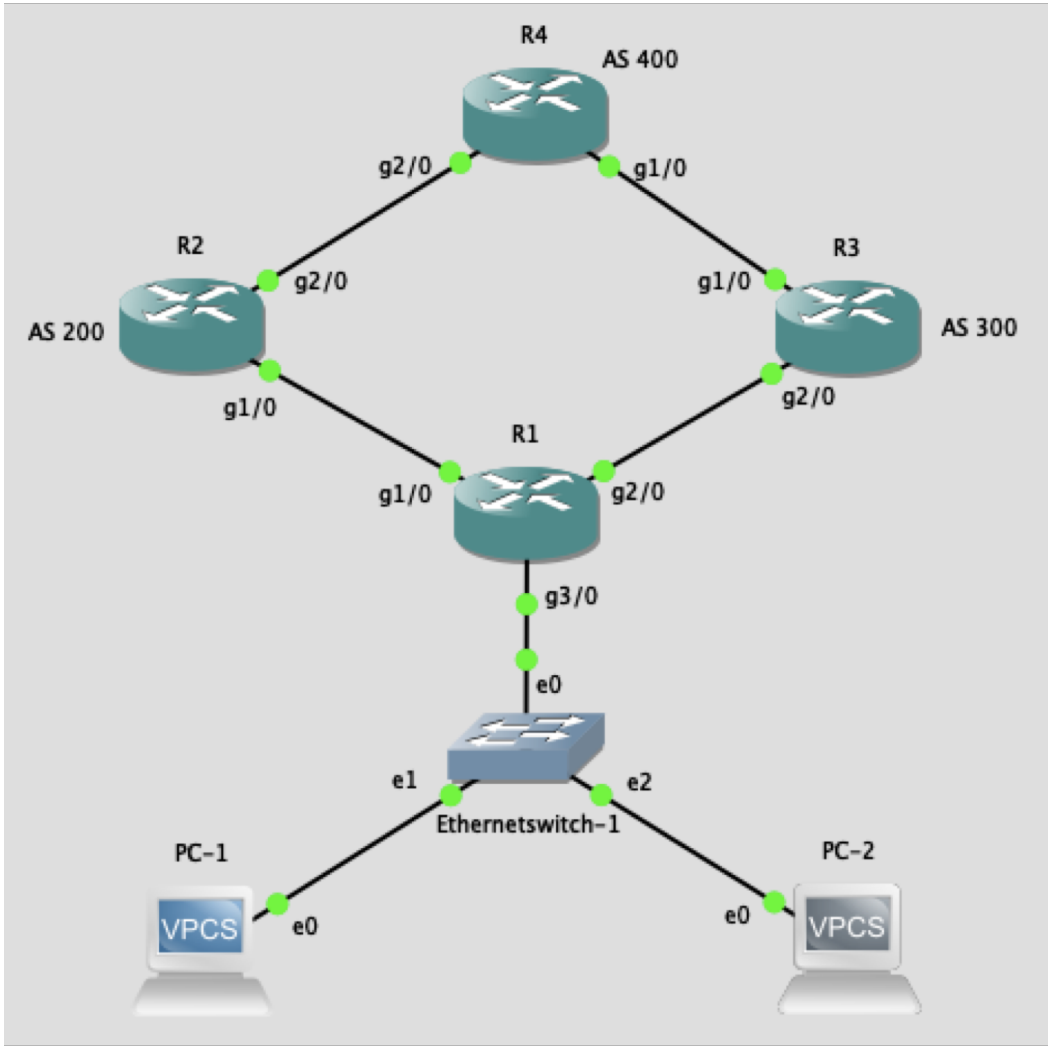
IPSLA and PBR

Lab Activity

Topology


R4
AS 400
g2/0
g1/0
R2
R3
g2/0
g1/0
AS 200
AS 300
g1/0
g2/0
R1
g1/0
g2/0
g3/0
e0
e1
e2
Ethernetswitch-1
PC-1
PC-2
VPCS
VPCS
e0
e0

Network and IP Plan

- IP Plan
  - Peering: 100.100.XY.X(Y)/24
  - Loopback 10 in R4: 50.50.50.1/24
- R1:
  - Gi3/0.10: Vlan 10, IP: 10.10.10.1/24
  - Gi3/0.20: Vlan 20, IP: 10.10.20.1/24
- Ethernetswitch-1 configuration:
  - Port 0: dot1q
  - Port 1: access vlan 10
  - Port 2: access vlan 20
- VPCS:
  - PC1: 10.10.10.10/24 GW: 10.10.10.1
  - PC1: 10.10.20.20/24 GW: 10.10.20.1

Task 1: Basic Configuration

Task 1.1: Interface Configuration

- Configure all routers
  - Loopback
  - Interface IP

Example: R1

```
interface GigabitEthernet1/0
 description Connected to R2 Gi1/0
 ip address 100.100.12.1 255.255.255.0
 no shutdown
!
interface GigabitEthernet2/0
 description Connected to R3 Gi2/0
 ip address 100.100.13.1 255.255.255.0
 no shutdown
!
interface GigabitEthernet3/0
 description Connected LAN Switch
 no ip address
 no shutdown
!
interface GigabitEthernet3/0.10
 encapsulation dot1Q 10
 description Connected to VLAN 10
 ip address 10.10.10.1 255.255.255.0
 no shutdown
!
interface GigabitEthernet3/0.20
 encapsulation dot1Q 20
 description Connected to VLAN 20
 ip address 10.10.20.1 255.255.255.0
 no shutdown
```

Task 1.2: Ethernetswitch-1 Configuration

Task 1.2: Ethernetswitch-1 Configuration

- Configure the Ethernetswitch-1
  - Port 0: dot1q
  - Port 1: access vlan 10
  - Port 2: access vlan 20

Task 1.3: VPCS Configuration

Example: VPCS-1

```
VPCS-1> ip 10.10.10.10 255.255.255.0 10.10.10.1

VPCS-1> save
```

Verify:

```
VPCS-1> show ip

NAME        : VPCS-1[1]
IP/MASK     : 10.10.10.10/24
GATEWAY     : 10.10.10.1
DNS         :
MAC         : 00:50:79:66:68:00
LPORT       : 10038
RHOST:PORT  : 127.0.0.1:10039
MTU         : 1500
```

Task 2: Default Route Configuration

Task 2: Default Route Configuration

- Configure default route in R1
  - Towards R2 with default AD
  - Towards R3 with AD = 10

Example: R1

```
ip route 0.0.0.0 0.0.0.0 100.100.12.2
ip route 0.0.0.0 0.0.0.0 100.100.13.3 10
```

Task 3: BGP Configuration

Task 2: BGP Configuration

- Configure BGP
  - Between R2-R4 and R3-R4
  - Advertise network
  - Originate default route in R4

Example: R4

```
router bgp 400
 neighbor 100.100.24.2 remote-as 200
 neighbor 100.100.34.3 remote-as 300
 address-family ipv4
  network 50.50.50.0 mask 255.255.255.0
  neighbor 100.100.24.2 default-originate
  neighbor 100.100.34.3 default-originate
```

Task 4: NAT Configuration

Task 4: NAT Configuration

- Configure NAT in R1
  - Interface NAT with overload (PAT)
  - Outside: Gi1/0 and Gi2/0
  - Inside: Gi3/0.10 and Gi3/0.20

Example: R1

```
access-list 1 permit 10.10.10.0 0.0.0.255
access-list 2 permit 10.10.20.0 0.0.0.255
!
ip nat inside source list 1 interface GigabitEthernet1/0 overload
ip nat inside source list 2 interface GigabitEthernet2/0 overload
!
int gi3/0.10
 ip nat inside
!
int gi3/0.20
 ip nat inside
!
int gi1/0
 ip nat outside
!
int gi2/0
 ip nat outside
```

Routing Issue

- Check R1's routing table for the default route

```
R1# show ip route
S*    0.0.0.0/0 [1/0] via 100.100.12.2
```

- Ping and trace R4's loopback from the VPCS-1 and VPCS-2.
- Shutdown R2's Gi2/0 interface.

```
R2(config)# int gi2/0
R2(config-if)# shutdown
```

- Check R1's routing table for the default route

```
R1# show ip route
S*    0.0.0.0/0 [1/0] via 100.100.12.2
```

- Ping and trace R4's loopback from the VPCS-1 and VPCS-2.
- Restore R2's Gi2/0 interface.

What’s the problem you see?

Can both LAN reach outside if the primary default path goes down?

Task 2: IP Service Level Agreement

IP SLA

- Define one or more IP SLAs operations (or probes).

  ip sla *operation_number*

- Define an ICMP echo operation from source to target.

  icmp-echo {*destination-ip-address*

  frequency *seconds*

- Schedule an IP SLA operation.

  ip sla schedule *operation-number* [life {forever | *seconds*}] [start-time {*hh:mm*[*:ss*] [*month day* | *day month*] | pending | now | after *hh:mm:ss*}] [ageout *seconds*] [recurring]*]*

- Define one or more tracking objects, to track the state of IOS IP SLAs operations.

  track *object-number* **ip sla** *operation-number* **{state | reachability}**

- Specify a period of time to delay communicating state changes of a tracked object.

  delay up *seconds* **down** *seconds*

- Define the action associated with the tracking object.

  ip route *prefix mask address* **track** *number*

Task 2: IP SLA

- Configure IP SLA in R1

```
ip sla 1
 icmp-echo 100.100.24.2
  frequency 5
!
ip sla schedule 1 life forever start-time now
track 5 ip sla 1 reachability
 delay down 6 up 3
!
no ip route 0.0.0.0 0.0.0.0 100.100.12.2

ip route 100.100.24.0 255.255.255.0 100.100.12.2
ip route 0.0.0.0 0.0.0.0 100.100.12.2 track 5
```

Task 2: IP SLA

- Configure IP SLA in R1
- Keep PING running to 50.50.50.1 in PC1 and PC2

```
PC-1> ping 50.50.50.1 -c 1000
```

- Check R1's routing table for the default route

```
R1# show ip route
S*    0.0.0.0/0 [1/0] via 100.100.12.2
```

- Ping and trace R4's loopback from the VPCS-1 and VPCS-2.
- Shutdown R2-R4 link.

```
R2(config)# int gi2/0
R2(config-if)# shutdown
```

- Check R1's log

```
%TRACKING-5-STATE: 1 ip sla 1 reachability Up->Down
```

- Check R1's routing table for the default route

```
R1# show ip route
S*    0.0.0.0/0 [10/0] via 100.100.13.3
```

- Ping and trace R4's loopback from the VPCS-1 and VPCS-2.

Task 2: IP SLA

- Bring up R2-R4 link.

```
R2(config)# int gi2/0
R2(config-if)# no shutdown
```

- Check R1's log

```
%TRACKING-5-STATE: 1 ip sla 1 reachability
DOWN->UP
```

- Check R1's routing table for the default route

```
R1# show ip route
S*    0.0.0.0/0 [1/0] via 100.100.12.2
```

Task 2: IP SLA

- Verification

  show ip sla configuration [*operation***]**

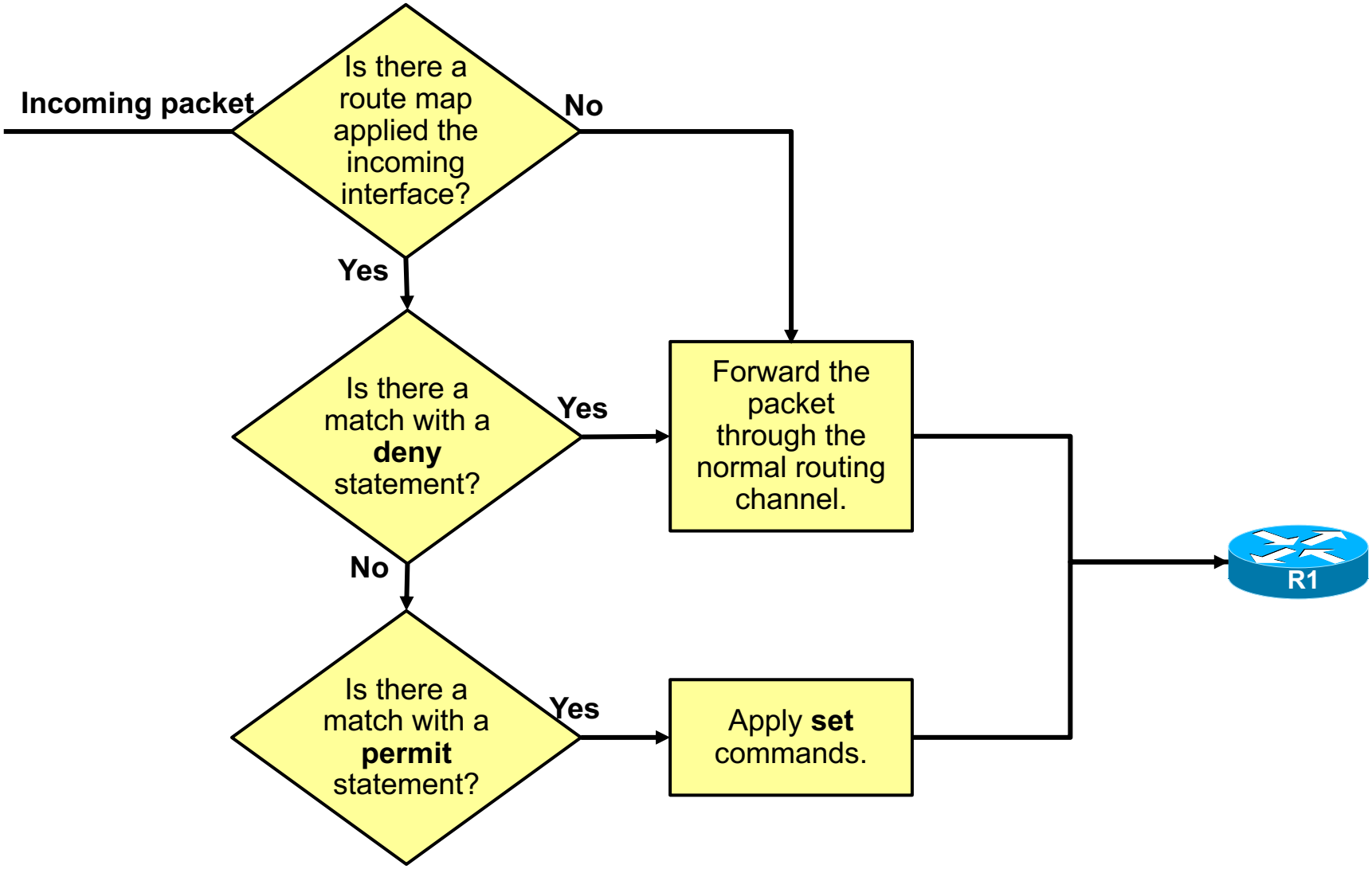
  show ip sla statistics [*operation-number* **| details]**

What’s the problem you see?

Can both uplinks get upload traffic simultaneously?

Task 3: Policy Based Routing (PBR)

Logical PBR Operation



PBR Configuration

```
Router(config)#
route-map map-tag [permit | deny] [sequence-number]
```

- Defines the route map conditions.

```
Router(config-route-map)#
match {conditions}
```

- Defines the conditions to match.

```
Router(config-route-map)#
set {actions}
```

- Defines the action to be taken on a match.

```
Router(config-if)#
ip policy route-map map-tag
```

- Apply the route-map to the incoming interface.

Task 3: PBR

- Check R1's routing table

  `show ip route`

- Check the trace repot
  - Path from VPCS-1 to Loopback 10 of R4
  - Path from VPCS-2 to Loopback 10 of R4

    `trace` *`ip_address`*

Task 3: PBR

- Configure PBR in R1
  - Configure ACL for the LAN IP Block
  - Configure Route-map
    - Match the ACL
    - Set next-hop
  - Configure the route-map in the ingress interface
- Check the trace path from VPCS-1 and VPCS-2

Example: R1

```
access-list 2 permit 10.10.20.0 0.0.0.255

route-map UPLOAD permit 10
 match ip address 2
 set ip default next-hop 100.100.13.3

int gi3/0.20
 ip policy route-map UPLOAD
```

Task 3: PBR

- Shutdown R2-R4 link.

```
R2(config)# int gi2/0
R2(config-if)# shutdown
```

- Check trace/ping to R4 from VPCS-1 and VPCS-2
  - Why it fails from VPCS-1 ?

Task 3: PBR

- Verification

  show ip policy

  show route-map [*map-name*]

Question?