IPv6

Lab Manual

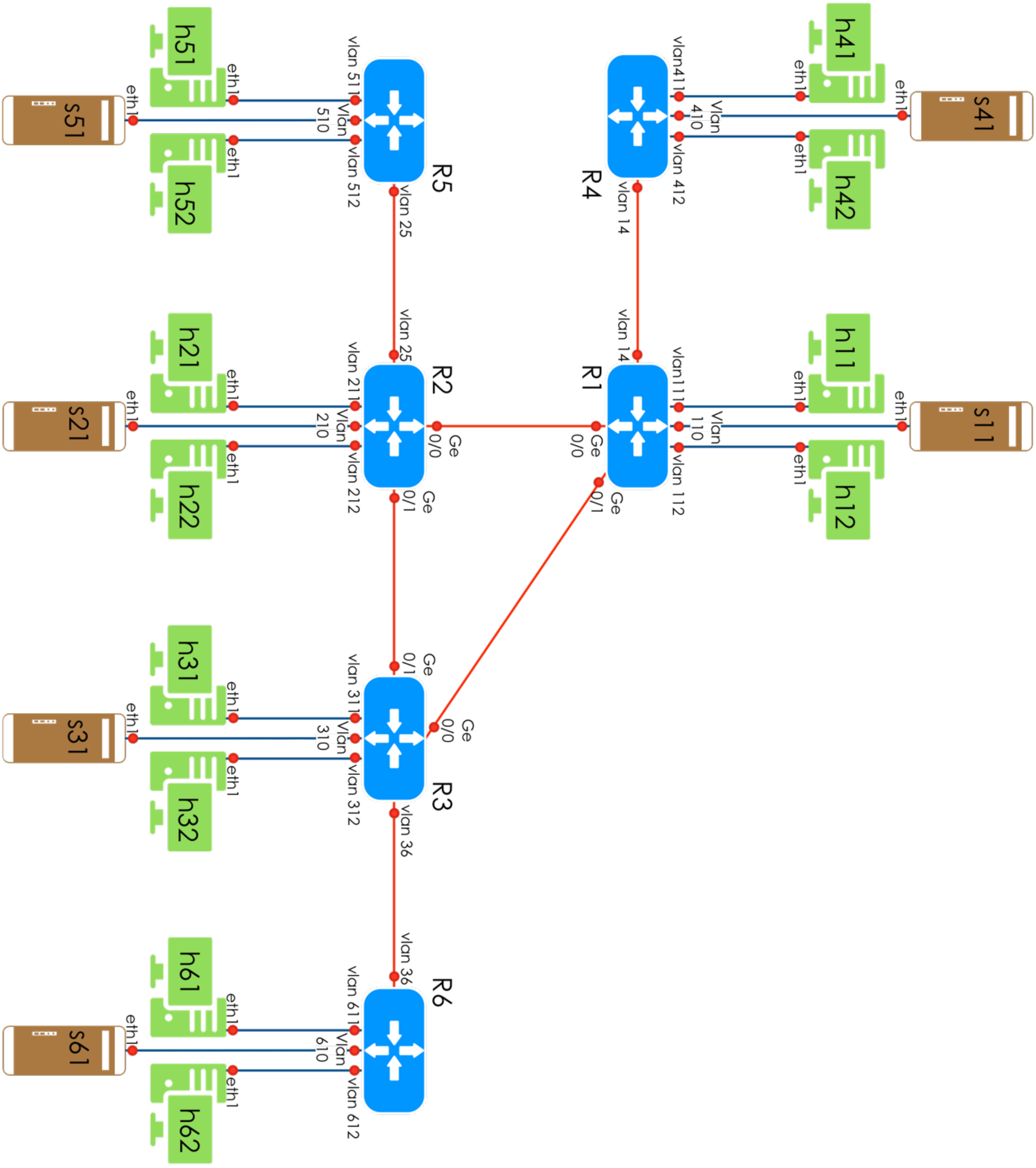
@AFRINICTRAINING

Solution Guide and Hints at: bit.ly/6lab-solution-guide

CONTENTS

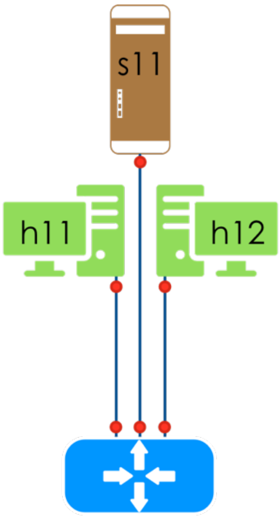
i. GENERAL TESTBED TOPOLOGY

Each testbed consists of 18 devices grouped into 6 pods (1 routers, 2 hosts and a server) as shown



Device	Interface	Connected to	Device	Interface	Connected to
R01	Ge 0/1	R2 [Ge 0/0]	R02	Ge 0/0	R1 [Ge 0/0]
	Ge 0/0	R3 [Ge 0/0]		Ge 0/1	R3 [Ge 0/1]
	vlan 14	R4 [vlan14]		vlan 25	R5 [vlan25]
	vlan 111	H11 [eth1]		vlan 211	H21 [eth1]
	vlan 112	H12 [eth1]		vlan 212	H22 [eth1]
	vlan 110	S11 [eth1]		vlan 210	S21 [eth1]
R03	Ge 0/0	R1 [Ge 0/1]	R04	vlan 14	R1 [vlan 14]
	Ge 0/1	R2 [Ge 0/1]		vlan 411	H41 [eth1]
	vlan 36	R4 [vlan 36]		vlan 412	H42 [eth1]
	vlan 311	H31 [eth1]		vlan 410	S41 [eth1]
	vlan 312	H32 [eth1]			
	vlan 310	S31 [eth1]			
R05	vlan 25	R2 [vlan 25]	R06	vlan 36	R3 [vlan 36]
	vlan 511	H51 [eth1]		vlan 611	H61 [eth1]
	vlan 512	H52 [eth1]		vlan 612	H62 [eth1]
	vlan 510	S51 [eth1]		vlan 610	S61 [eth1]

During the workshop, the class is divided into groups, each is assigned to one testbed. Each group is will then organize themselves into 6 sub-groups, one per pod which is the basic unit around which all these exercises are built. A pod is illustrated below.



ii. IP ADDRESS BLOCKs FOR USE DURING EXERCISES

Testbed	IPv4 Prefix	IPv6 Prefix
Mauritius 01	10.10.0.0/16	2001:db8:10::/48
Johannesburg	10.20.0.0/16	2001:db8:20::/48
Mauritius 02	10.30.0.0/16	2001:db8:30::/48
Cape Town	10.40.0.0/16	2001:db8:40::/48

Replacing X with 10 for Mauritius 1(mru1), 20 for Johannesburg(jnb), 30 for Mauritius 2(mru2), and 40 for Cape Town(cpt), the generic addressing plan for each testbed is as follows:

Router	Interface	IPv4 address	IPv6 address	Connected to	IPv4 address	IPv6 address
1	Ge 0/0	10.X.0.1/30	2001:db8:X:12::2/127	R2[Ge0/0]	10.X.0.2/30	2001:db8:X:12::3/127
	Ge 0/1	10.X.0.5/30	2001:db8:X:13::2/127	R3[Ge0/0]	10.X.0.6/30	2001:db8:X:13::3/127
	vlan 14	10.X.0.9/30	2001:db8:X:14::2/127	R4[vlan14]	10.X.0.10/30	2001:db8:X:14::3/127
	vlan 111	10.X.11.1/24	2001:db8:X:111::1/64	H11[eth1]	10.X.11.2/24	2001:db8:X:111::2/64
	vlan 112	10.X.12.1/24	2001:db8:X:112::1/64	H12[eth1]	10.X.12.2/24	2001:db8:X:112::2/64
	vlan 110	10.X.15.1/24	2001:db8:X:151::1/64	S11[eth1]	10.X.15.2/24	2001:db8:X:151::2/64
	lo 1	10.X.1.1/32	2001:db8:X:11::1/128			
2	Ge 0/0	10.X.0.2/30	2001:db8:X:12::3/127	R1[Ge0/0]	10.X.0.1/30	2001:db8:X:12::2/127
	Ge 0/1	10.X.0.13/30	2001:db8:X:23::2/127	R3[Ge0/1]	10.X.0.14/30	2001:db8:X:23::3/127
	vlan 25	10.X.0.17/30	2001:db8:X:25::2/127	R5[vlan25]	10.X.0.18/30	2001:db8:X:25::3/127
	vlan 211	10.X.21.1/24	2001:db8:X:221::1/64	H21[eth1]	10.X.21.2/24	2001:db8:X:221::2/64
	vlan 212	10.X.22.1/24	2001:db8:X:222::1/64	H22[eth1]	10.X.22.2/24	2001:db8:X:222::2/64
	vlan 210	10.X.25.1/24	2001:db8:X:251::1/64	S21[eth1]	10.X.25.2/24	2001:db8:X:251::2/64
	lo 1	10.X.1.2/32	2001:db8:X:22::1/128			
3	Ge 0/0	10.X.0.6/30	2001:db8:X:13::3/127	R1[Ge0/1]	10.X.0.6/30	2001:db8:X:13::2/127

	Ge 0/1	10.X.0.14/30	2001:db8:X:23::3/127	R2[Ge0/1]	10.X.0.14/30	2001:db8:X:23::2/127
	vlan 36	10.X.0.21/30	2001:db8:X:36::2/127	R6[vlan36]	10.X.0.21/30	2001:db8:X:36::3/127
	vlan 311	10.X.31.1/24	2001:db8:X:331::1/64	H31[eth1]	10.X.31.2/24	2001:db8:X:331::2/64
	vlan 312	10.X.32.1/24	2001:db8:X:332::1/64	H32[eth1]	10.X.32.2/24	2001:db8:X:332::2/64
	vlan 310	10.X.35.1/24	2001:db8:X:351::1/64	S31[eth1]	10.X.35.2/24	2001:db8:X:351::2/64
	lo 1	10.X.1.3/32	2001:db8:X:33::1/128			
4	vlan 14	10.X.0.10/30	2001:db8:X:14::3/127	R1[vlan14]	10.X.0.9/30	2001:db8:X:14::2/127
	vlan 411	10.X.41.1/24	2001:db8:X:441::1/64	H41[eth1]	10.X.41.2/24	2001:db8:X:441::2/64
	vlan 412	10.X.42.1/24	2001:db8:X:442::1/64	H42[eth1]	10.X.42.2/24	2001:db8:X:442::2/64
	vlan 410	10.X.45.1/24	2001:db8:X:451::1/64	S41[eth1]	10.X.45.2/24	2001:db8:X:451::2/64
	lo 1	10.X.1.4/32	2001:db8:X:44::1/128			
5	vlan 25	10. X.0.18/30	2001:db8:X:25::3/127	R2[vlan25]	10.X.0.17/30	2001:db8:X:25::2/127
	vlan 511	10.X.51.1/24	2001:db8:X:551::1/64	H51[eth1]	10.X.51.2/24	2001:db8:X:551::2/64
	vlan 512	10.X.52.1/24	2001:db8:X:552::1/64	H52[eth1]	10.X.52.2/24	2001:db8:X:552::2/64
	vlan 510	10.X.55.1/24	2001:db8:X:555::1/64	S51[eth1]	10.X.55.2/24	2001:db8:X:555::2/64
	lo 1	10.X.1.5/32	2001:db8:X:55::1/128			
6	vlan 36	10.X.0.22/30	2001:db8:X:36::3/127	R3[vlan36]	10.X.0.21/30	2001:db8:X:36::2/127
	vlan 611	10.X.61.1/24	2001:db8:X:661::1/64	H61[eth1]	10.X.61.2/24	2001:db8:X:661::2/64
	vlan 612	10.X.62.1/24	2001:db8:X:662::1/64	H62[eth1]	10.X.62.2/24	2001:db8:X:662::2/64
	vlan 610	10.X.65.1/24	2001:db8:X:651::1/64	S61[eth1]	10.X.65.2/24	2001:db8:X:651::2/64
	lo 1	10.X.1.6/32	2001:db8:X:66::1/128			

iii. DNS INFORMATION FOR USE DURING EXERCISES

The DNS domain for each pod is "<node_name>.pod<R>.local" where

- <node_name> is the name of the node (host or router) as appears in topology diagram
- R is in the range [1 → 6] i.e. the router number
- Thus h21.pod2.local is the fully qualified domain name
  - The second host connected to router 2 (h21)
  - Of pod #2 (pod2) i.e. the pod built around Router 2

iv. DEVICE PRE-CONFIGURATIONS

All nodes	• Hostname: as shown in general topology diagram • Login username: 6deploy • Login password: 6deploy • IPv4 addressing: already configured as per section 1.2 • Management interface: eth1
Host VMs	• IPv4 default gateway: IPv4 address of directly connected router interface • Operating System: Ubuntu 16.04 LTS
Server VMs	• Configured as DNS server using Bind • Domain: pod<R>.local • Full working IPv4 zones • Accepts dynamic DNS registrations from DHCP service • Configured as DHCP server • Dynamically registers allocated addresses with the DNS service
Routers	• Management interface: vlan99 • Routing • Runs multi-area OSPFv2 as per topology on page 10 • All host interfaces configured as passive in routing protocols • BGP-pre-configuration saved in NVRAM. Only load when instructed to for BGP lab • Router loopbacks (lo0) are carried in OSPFv2 • Host LAN prefixes are announced via iBGP • eBGP enabled between all hub routers (R01, R02 & R03) announcing their own prefixes as well as directly connected spoke routers

v. HOW TO ACCESS AND USE DEVICES

Management interfaces (eth1 for hosts and vlan99 for routers) are configured with globally routable IPv4 addresses. Proceed thus to log in:

```
ssh -l 6deploy <host_name>.<testbed>.6lab.afrinic.net
```

- <host_name> = device hostname e.g. r01 - r06, h11 etc.
- <testbed_id> = mru1, mru2, cpt and jnb

For example, to log into the second host connected to the router 01 in Cape Town, we'd type

```
ssh -l 6deploy h21.cpt.6lab.afrinic.net --or–
ssh 6deploy@h21.cpt.6lab.afrinic.net
```

vi. LAB RULES

Unless explicitly asked to by a trainer, violating any of these will cut you off from your device.

1. Don't change the management interface
   - vlan 99 on routers
   - eth0 on hosts
2. Don't remove IPv4 static routes.
3. Don't change login/password information or filtering policy

When in doubt, ask one of the trainers.

1. ENABLE IPv6 ROUTING & CONFIGURE STATIC ADDRESSES

Objective: Enable and verify IPv6 routing on the router and manually configure IP addresses on all non-management interfaces of all devices.

(a) Verify that IPv6 routing is enabled on the router, if it is not, enable it.

(b) Verify that IPv6 is enabled on each host and server. if it is not, enable it.

(c) Without configuring any addresses, enable IPv6 on the all interfaces of your router.

(d) Still without explicitly configuring any addresses, test IPv6 connectivity to all directly connected devices.

(e) Configure IPv6 addresses on your devices according to the address plan provided. (be careful not to lock yourself out by modifying the management interfaces)

(f) Configure the appropriate default routers on each of the hosts and server.

(g) Test reachability to all devices on the new addresses configured in (e).

2. CONFIGURE AND VERIFY BASIC DNS IN IPv6

Objective: Configure the DNS service (Bind) so that hosts' static IP addresses (configured in 2.1 above) can be reached by their fully qualified domain names. (The Bind service has already been installed, configured and tested)

(a) Configure the hosts and server to point to the server as their resolver

(b) On any host, verify that a DNS service exists and is ready to receive name resolution requests on the server.

(c) From any host, ensure that the server can resolve the IPv4 address of the other host and server

(d) Insert AAAA records for the hosts' static IPv6 addresses in the server's forward zone file. The zone file is located on your server at </etc/bind/db.podX>

(e) Reload the Bind service to ensure that the zone file modifications are correct (fix any errors)

(f) Verify that you can resolve host1 and host2 by names from each other

(g) Test connectivity between hosts with their names

3. CONFIGURE AND VERIFY SLAAC

Objective: Using SLAAC, provision IPv6 configuration to the hosts from the router.

(a) Configure the router to advertise a prefix out each interface on which hosts are connected as follows

- Prefix **fdX:c001:51ac:hh::/64** where:
- X = as defined in section 1.2
- hh is the host number e.g host with hostname h51, has hh = 51 (first host off R5)
- Host must NOT need their default gateway to reach other hosts in the same subnet

(b) Configure the hosts to automatically generate an address from the information sent by the router.

(c) On each host display the IPv6 interface information and note

- ○ How many non-link-local addresses the interface has
- ○ List the non-link-local addresses

(d) Configure the router **not to advertise** the prefix of the manual address configured in exercise 1(e) in RA messages.

(e) Restart the networking service on the hosts

(f) Repeat step (c) and note the difference.

4. CONFIGURE AND VERIFY STATELESS DHCPv6

Objective: Configure the hosts request and use DNS information from the server.

(a) Ping the other host by its FQDN. Was the ping successful? (why/why not?)

(b) On the server, configure the DHCPv6 service as follows:

- ○ The DNS server should be the IPv6 address of the server
- ○ The domain search list should be "pod<**R**>.local" where "R" is the router number.

(c) Configure the router as follows

- Inform the hosts that further configuration information should be gotten through DHCPv6
- Forward all DHCPv6 messages on to the server

(d) Reload the network service on both hosts so they refresh their configurations

(e) Redo exercise (a) above. Why was it successful this time?

5. CONFIGURE AND VERIFY STATEFUL DHCPv6

Objective: Configure the hosts to get addresses automatically from the server using DHCPv6. Before starting this exercise, remove all the global IPv6 addresses configured on the Routers Interfaces connected to the hosts HX1 and HX2 only.

(a) On the server, configure two DHCPv6 scopes with the following specifications: (hint: the DHCPv6 configuration file is "/etc/dhcp/dhcpd6.conf"

- ○ Addresses should be handed out from the prefix **fdX:c001:d4c:hh::/64** where the variables "X" and "hh" are as previously defined.
- ○ The DNS server should be the IPv6 address of the server
- ○ The domain search list should be "pod<**R**>.local" where "R" is the router number.

(b) On the router, configure the interfaces to the hosts as follows:

- ○ Configure an address from the prefix **fdX:c001:d4c:hh::/64** (use a suitable interfaceID)
- ○ The router should inform the hosts to use DHCPv6 for address configuration
- ○ Hosts must NOT need their default gateway to reach other hosts in the same subnet
- ○ The router should forward DHCP messages from hosts to the server.

(c) Configure the hosts to request and use configuration information through DHCPv6

(d) From your host, find out and list the addresses that were given to the other hosts in your pod

(e) Display the lease database on the server.

- What addresses did it hand out to the hosts?
- Are they the same as those addresses from (d) above?

(f) Write down all the addresses on the host's interface, what is their prefix length?

6. VERIFY IPv6 ADDRESS SELECTION

Objective: Observe the IPv6 address selection logic when multiple addresses exist on an interface.

(a) From your first host, ping

- The router's link local address
- The router's directly connected non-link-local addresses
- The other host in your pod.

(b) Examine the forwarding table on your host and write down which of the host's addresses was used for each of the destinations above. What do you notice?

7. CONFIGURE AND VERIFY IPv6 PREFIX DELEGATION

Objective: Configure the server to provide **prefixes** to the router for use on its local networks, then automatically use these prefixes to provision addressing on the host LANs.

(a) Configure the server to hand out /56 prefixes from the **fdX:c00R:b10c::/48** (X and R are defined previously)

(b) Configure the router to request a prefix from the server and store it as "DelegatedPrefix"

(c) On the server, find out which prefix has been handed out to the router.

(d) On the router, find out what prefix was received from the server. How is this different from the prefix from (c) above?

(e) Configure the router to automatically assign two /64 blocks from the delegated prefix to each of the host interface.

(f) On the router, list the non-link-local addresses for the host-facing interfaces, how are they related to the delegated prefix?

(g) Configure both hosts to use SLAAC.

(h) List the addresses that the hosts now have

(i) From each host, find out and write the addresses that were given to the other hosts in your pod

(j) Verify IP connectivity between all hosts using their PD-delegated addresses

8. CONFIGURE AND VERIFY BASIC IPv6 ACCESS LISTS

Objective: Limit management plane access on routers to specific hosts.

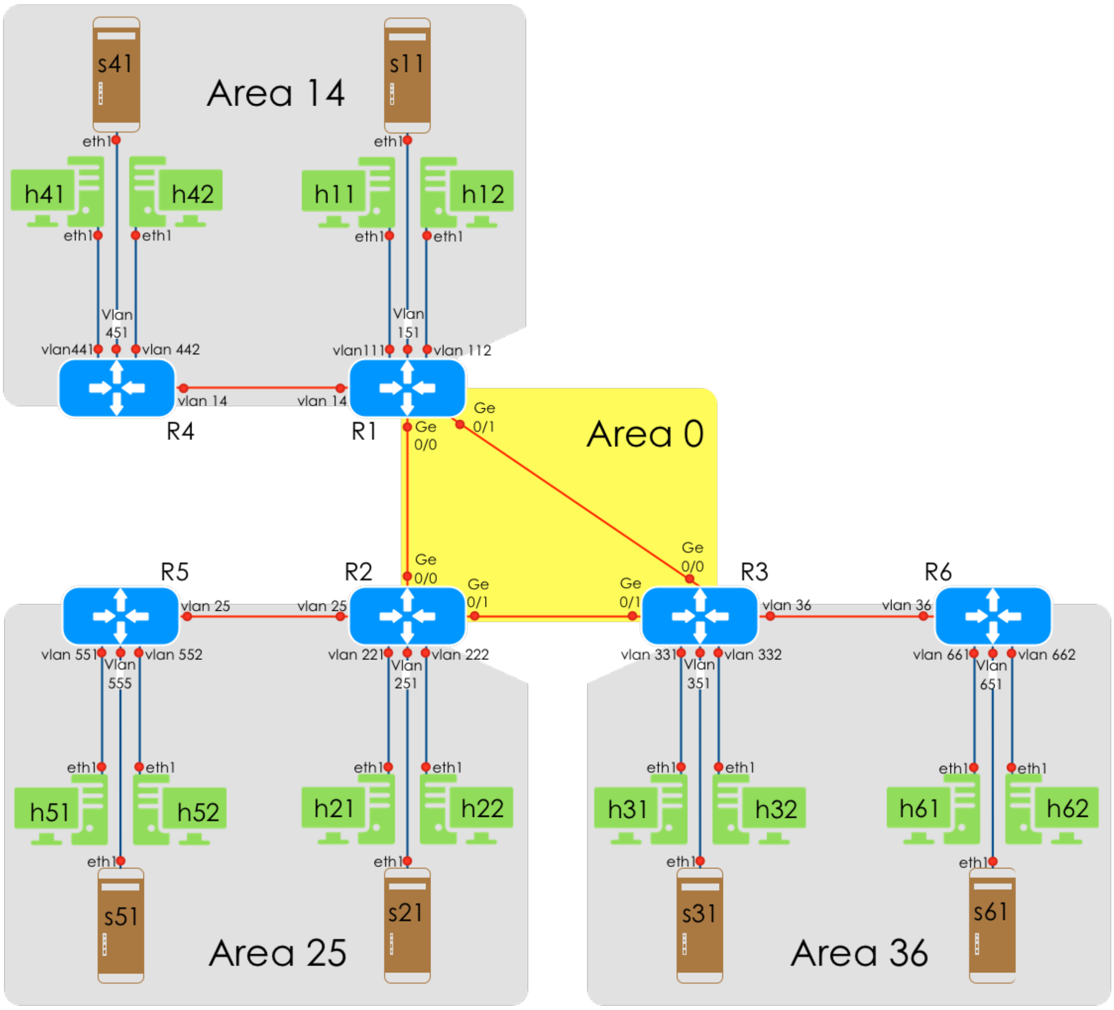
(a) Verify that from each host, you can telnet/ssh to the routers and servers.

(b) Configure the router to only allow SSH access to the server on its static IPv6 address and only from host1's static IPv6 address.

(c) Configure the router to only allow FTP access to the server on its static IPv6 address and only from host2's static IPv6 address.

9. CONFIGURE AND VERIFY OSPFv3 INTRA-DOMAIN ROUTING

Objective: Configure and verify dual stack multi-area intra-domain routing using OSPF as specified in the diagram below. (Note that OSPFv2 has already been pre-configured)



(a) Configure OSPFv3 according to the topology above. OSPF packets should NOT be sent on host/server interfaces.

(b) Redistribute the loopback interfaces of the routers into OSPF.

(c) How many different OSPFv3 routes are there in each router and what are their types?

(d) Display, compare and contrast the IPv4 and IPv6 OSPF topology tables. Which LSA types are new to IPv6?)

(e) Test IPv4 and IPv6 reachability to every host and server in every other pod in your testbed

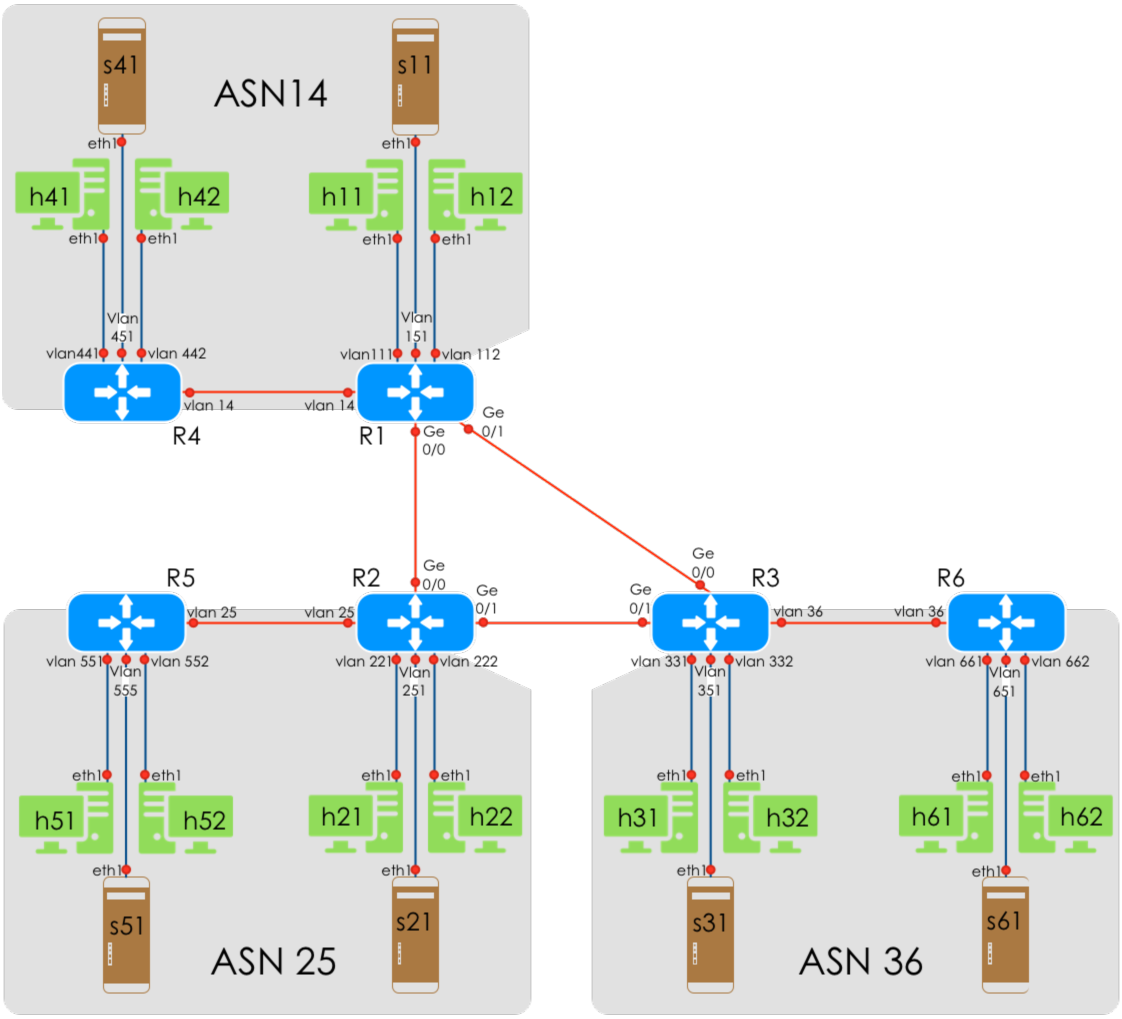
(f) Save the configuration of your router to flash with the name <city>-r0**x**-ospf23.cfg (where <city> is the name of the city where the training is taking place and "x" is the router number). **You'll need this configuration later.**

(g) Disable the OSPFv2 configuration from your router.

(h) Verify that you can still reach (via ping) the loopback interfaces of the other routers over IPv6 but not over IPv4. *(What does this tell you about the dependence of IPv4 and IPv6?)*

10. CONFIGURE AND VERIFY BGP FOR IPv6

Objective: Configure and verify inter-domain routing using MP-BGP and exchange client prefixes.



(a) On your router, load the configuration called bgp-pre.cfg from flash

(b) Verify that:

- Only loopbacks within the same ASN are carried in OSPFv3
- Every node in every ASN is currently reachable over IPv4 through BGP
- Nodes in other pods are not currently reachable over IPv6

(c) Create an iBGP session (over IPv6) between the PE and CE in your Autonomous System.

(d) Advertise host/server prefixes to other peers in the AS.

(e) On each PE router, create an eBGP session (over IPv6) to neighbouring PE routers.

(f) The PE should advertise all client prefixes learnt via iBGP as well as its own prefixes to other ASNs

(g) Verify that there is full reachability over IPv6 across the entire testbed.

11. CONFIGURE AND VERIFY MANUAL TUNNELS

Objective: Connect IPv6 islands over an IPv4-only network using manual 6in4 tunnels.

(a) Remove all IPv6 addresses from all inter-connecting links between your pod and the neighbouring pods. (there is thus no native IPv6 between the pods)

This removes native IPv6 connectivity between the pods. The OSPFv3 and IPv6 e-BGP sessions should now fail. You now have islands of dual-stacked pods linked by IPv4 only.

(b) Verify that you can no longer reach hosts and servers in other pods over IPv6.

(c) Create manual 6in4 tunnels between your pod and the neighbouring pods.

- Address these tunnels with IPv6 addresses from the address plan given in 1.2.
- Source and terminate the tunnels on the loopback interfaces.

(d) Modify the OSPFv3 and eBGP session to use the newly created tunnels for their IPv6 sessions.

(e) Verify that you can now reach the all other hosts in all other pods over IPv6.

12. CONFIGURE AND VERIFY 6RD TUNNELS

Objective: Configure and verify 6rd automatic tunnelling mechanism to connect dual-stacked (or IPv6-only) networks over an IPv4-only network. Treat each PE-CE pair an individual 6rd domain, with the PE router (1, 2, & 3) as a border relay (BR) and the CE (4, 5, 6) as the 6rd CPE. This transforms the PE router into a 6rd BR and the CE into a 6rd CPE.

(a) On both the PE and CE load the dual-stacked OSPF configuration you saved in task 9 (f).

(b) On the PE, creating a 6rd virtual interface (a tunnel) with the following specifications:

- Tunnel source: the interface that connects to the CE.
- 6rd prefix: fd**vw:xy00::/32** (where w,v,x,y are as defined previously.
- IPv4MaskLen: 24.
- IPv6 enabled

(c) Create a static route to direct all traffic destined to the 6rd prefix to the 6rd virtual interface.

(d) Redistribute the 6rd Prefix into the OSPFv3 routing protocol so it gets propagated to other 6rd domains.

(e) On the CE, create an IPv6 general prefix with the following specs:

- Name: "6rdDelegatedPrefix"
- Type: "6rd"
- Tunnel interface: Tunnelxy (x and y are as defined previously)

(a) On the CE create a 6rd virtual interface with the following specifications:

- Tunnel source: interface that connects to the PE.
- 6rd prefix: same as specified for the BR
- IPv4MaskLen of 24.
- BRIPv4Address: the IPv4 address on the PE-end of the CE-PE link.
- IPv6 enabled

(b) Direct traffic to use the 6rd virtual interface by creating the following IPv6 routes:

- A null route for the 6rd delegated prefix on this CPE.
- A static route for 6rd prefix to use the 6rd virtual interface created above as egress interface.
- A default route that points the subnet anycast address of the BR's 6rd delegated prefix.

(c) Verify that the 6rd tunnel works by pinging any IPv6 address on the remote end. (Note what happens when you ping a native IPv6 address vs a 6rd one)

(f) On the CE, configure each host interface as follows

- IPv6 address: from a unique subnet of the delegated prefix
- Hosts connected to that interface should use the prefix to automatically generate their own addresses