

## Lab 3 Postlab: BGP

Or: iBGP, eBGP, policy-options, and route redistribution

### What you will do:

1. Configure iBGP for peers in the same AS
2. Configure eBGP for peers in different AS's
3. Configure BGP to ensure proper operation across multiple AS's
4. Create and apply policies to achieve route redistribution from several different protocols
5. Create and apply policies to achieve route redistribution into several different protocols
6. Save configs & enter your answers for each task on Blackboard

### Things that you will need to know or learn:

1. Everything you learned about BGP in NET3008
2. Topology for the MySRLab lab as used for this lab (see next page)
3. CLI commands for configuring ports, interfaces, and the system interface (previous labs)
4. CLI commands for configuring OSPF (previous labs)
5. CLI commands for configuring BGP (see later page)

### What you need to submit and when:

1. Complete the "Lab 3 Post-lab" exercise on Blackboard with answers from your work, **before** your next lab period.

### Required Equipment:

- Credentials for booking a MySRLab session
- An actual MySRLab booking
- A PC with internet access, Java, and terminal program; eg. Putty (provided by you)

### Marks:

This post lab is worth 2/3 of the mark for Lab 3.

10% of your final mark is for labs done during the course of the semester.

### References and Resources:

- Your textbook and all your notes on BGP from NET3008 – Advanced Routing
- Command references provided in previous labs
- BGP configuration commands provided in this lab (below)
- MySRLab: remote-access lab facility hosted at Nokia's Kanata campus
- 7750 SR OS Routing Protocols Guide Ver 12.0, see the sections *BGP Configuration Process Overview* (pp. 685-686 in Ver 12.0), and *BGP Configuration Overview* through to *BGP Command Reference* (pp. 690-710 in Ver 12.0)

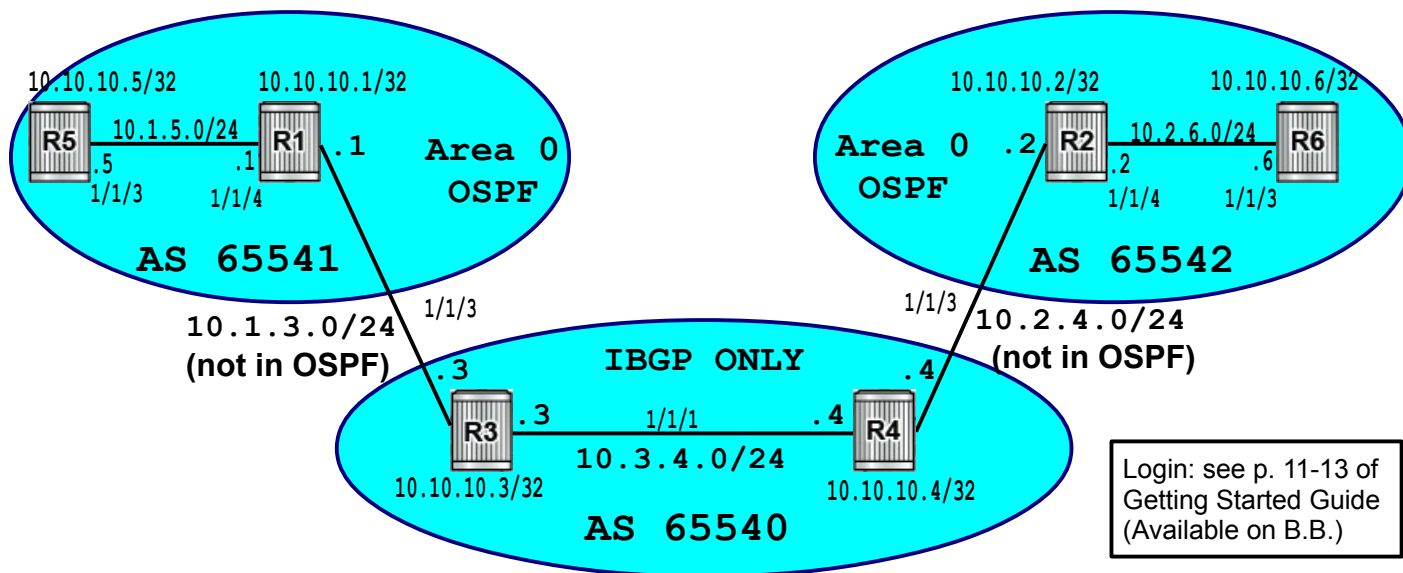
Tip: use google to search the terms:

7750 SR OS routing protocols guide 12.0 site:infoproducts.alcatel-lucent.com

**PLEASE see updated versions of either "Lab01-Prelab" or "Lab01-IntroToSR7750-CLI" for changes to ftp directory structure for saving & restoring config files!**

## Topology

This is the MySRLab network topology that is used for this lab. It is similar to the top half of the EDU topology from the weekly in-lab. **NOTE:** that **only some** links are configured and activated!



## Addressing Table for MySRLab Practice Labs

**ALL** routers are IPv4 addresses in the range **192.168.X.Y** with **X.Y** as given in the table below.

	PL3	PL4	PL5	PL6	PL7	PL8
R1	188.4	188.36	188.68	188.100	188.132	188.164
R2	188.5	188.37	188.69	188.101	188.133	188.165
R3	188.6	188.38	188.70	188.102	188.134	188.166
R4	188.7	188.39	188.71	188.103	188.135	188.167
R5	188.8	188.40	188.72	188.104	188.136	188.168
R6	188.9	188.41	188.73	188.105	188.137	188.169
PC1	188.10	188.42	188.74	188.106	188.138	188.170

The Ixia tester PC is accessed via the Remote Desktop Protocol (RDP), with clients available for Windows (built-in as "mstsc.exe"), Linux (rdesktop), and OSX (<http://cord.sourceforge.net/>)

Practice Lab	User ID	Password
PL3	See pages 4-5 of the <i>MySRLab Technical Primer</i> and page 8 of the <i>MySRLab Getting Started Guide</i> (available on Blackboard) for connectivity & login details.	
PL4		
PL5		
PL6		
PL7		
PL8		



## Command Reference

configure • router • autonomous-system {AS-#}

configure • router • bgp

configure • router • bgp • export <policy-stmt>

configure • router • bgp • group <group-id>

configure • router • bgp • group <group-id> • neighbor {IP-address}

configure • router • bgp • group <group-id> • neighbor {IP-address} • peer-as {AS-#}

configure • router • bgp • group <group-id> • next-hop-self

entry {#} • from protocol {direct | ospf | bgp | isis | rip} # a partial list of the protocol options

show • router • bgp • neighbor

show • router • bgp • routes

show • router • bgp • routes X.X.X.X/YY detail # to see all details of a specific route

environment • no more

---

### Task 1: Configure IP addresses, and OSPF areas in AS 65541 / 2

Configure **two** OSPF backbone areas, one in each AS in the right and left pods. To ensure your results match the expected answers, do **not** include the links joining AS's in the OSPF areas. Test to ensure that you have full reachability within each area.

### Task 2: Configure iBGP peers R3 – R4

In this task, you'll configure simple iBGP peering between R3 and R4. BGP peers are organized into (named) groups (remind you of OSPF or EIGRP areas?). Within a group, the individual peers are enumerated and configured.

Step 1. First, check how many routes appear in the routing tables of R3 and R4 **before** you configure BGP . You'll need to compare before-and-after.

Step 2. Within **configure router**, set the local AS number (65540) for the two routers.

Step 3. Based on the command reference (above), complete the remaining BGP configuration within **configure router bgp**: group name, peer address, and peer AS.

Step 4. If you've done everything correctly, the peering should happen immediately, but don't forget to wait a moment or two if things don't appear to work instantly.

Step 5. Check that the configuration and peering has been successful: use an appropriate **show** command to see the neighbours. (If not, do some troubleshooting!)

**Capture** the output for answering questions below.

Step 6. Check the routing table and compare with step 1. How many new routes?

Questions to answer:

- How many **new** routes now appear in the routing table after BGP peering?
- In the neighbour information you captured, look for the items "IPv4 Recd. Prefixes" and "IPv4 Active Prefixes". How many routes did the neighbor share?

### Task 3: Configure policy statements (single entry)

If you read p. 686 in the SR OS Routing Guide (see references above), you'll find the statement "If no export route policy statements specified, then all best and used BGP routes are advertised and **non-BGP routes are not advertised**." What's missing for full reachability (eg. system interfaces) is an export policy (aka route redistribution).

In the last post-lab, we had a brief look at how the "**prefix-list**" statement can be used as a criteria in a policy-statement rule. Another available criteria is the "protocol" statement, eg:

```
entry 10
  from protocol {proto-name}
  action accept
```

You'll need to read the command reference (above) to figure out what protocol name encompasses the system and outward facing interfaces (eg. equivalent of Cisco's "**connected**" routes). ... Or, you could use help available right from the CLI !

- Step 1. On each iBGP peer, configure a policy-option statement (rule) so that each peer will *share with* it's neighbor the system and outward facing interfaces. Don't forget the requirement for matched **begin** and **end** statements.
- Step 2. Apply the policy statement using the command "**export**" within the BGP context.
- Step 3. Don't forget to wait! Everything is based on timers, so it may take a minute or so for the network to converge.
- Step 4. Confirm that extra routes now appear in each peer's routing table. If not, do some troubleshooting.
- Step 5. Verify the routes with some ping tests from R3: ping R4 system interface, R4 interface to R2, and finally R2. Did all pings work? If not, why not? What help does traceroute provide? **Capture** the output for answering questions below.
- Step 6. Just like the previous task, check the peering relationship between the routers. **Capture** the output for answering questions below.
- Step 7. Display the BGP route table using a **show** command. **Capture** the output for answering questions below.

Questions to answer:

- For Cisco, we can specify the term "connected" in a route redistribution statement. What is the equivalent term in SR OS?
- How many extra routes now appear in each peer's routing table after you applied the policy statement?
- Which ping(s) failed? Why?
- In the neighbour information you captured, look for the items "IPv4 Recd. Prefixes" and "IPv4 Active Prefixes". How many routes did the neighbor share? How many were **not** previously known locally?
- In the BGP route table, how many of the learned routes are marked as "?" for "incomplete"? Why? Hint: You should know this from NET3008!

## Task 4: Configure eBGP peers between AS's

Now you'll configure eBGP. Notice how it's basically *identical* to iBGP (for now). To follow best practices, create a second BGP group, on the central routers, that is dedicated to eBGP peers.

- Step 1. On routers R3-R4, configure a second BGP group. Add R1 / R2 as appropriate.
- Step 2. On routers R1-R2, do **just enough** configuration for successful BGP peering; don't forget the AS number (Pod 1 = 65541, Pod 2 = 65542).
- Step 3. If you've done everything correctly, the peering should happen immediately, but don't forget to wait a moment or two if things don't appear to work instantly.
- Step 4. Use a **show** command to confirm the peering relationship on **all four** routers. Troubleshoot as necessary. **Capture** the completed output for answering questions below.
- Step 5. Check each routers' routing table. **Capture** the output.

Questions to answer:

- For each of the four routers, how many BGP peers does it have?
- How many routes now appear on **each** of the routers? Is that what you expect?

## Task 5: Configure policy statements on Edge routers R1 and R2 (multiple entry)

Routes on the R1 / R2 router tables come from two different sources: directly connected interfaces and OSPF. A suitable policy statement for exporting all possible routes will need **multiple** entries (well, two in this case). "Entry" statements operate in a logical "OR" manner: successful candidate routes may come from one entry statement OR the other.

- Step 1. Configure policy statements on the two routers so that **both** directly connected and OSPF routes are *shared with* BGP neighbors.
- Step 2. Don't forget to wait! Everything is based on timers, so it may take a minute or so for the network to converge.
- Step 3. Check the routing table on **all four** routers. The appearance of additional routes on the inner R3-R4 routers is a good indicator of convergence. **Capture** the output.
- Step 4. Verify the routes by pinging on R3 and R4: eg. from R3, ping all three interfaces on R1, then ping both interfaces on R5. Did all pings work? If not, why not? Use traceroute!
- Step 5. On R3, show the BGP route table and examine it **carefully**. Determine which routes are exchanged via BGP but do **not** appear in the main routing table. How many are there? From which AS do these originate? Repeat this for R4 to determine the pattern. **Capture** the output.
- Step 6. To help understand, **show** the details of one of the BGP routes that does **not** appear in the main routing table on R3, eg. for R6. Examine it **carefully** and look for the information from the "Flags". What does that tell you? **Capture** the output.

Questions to answer:

- How many routes now appear on **each** router? Is that what you expect?
- When you verified using pings from R3 (or R4), which pings failed? Why?
- On R3, how many routes are exchanged via BGP but do **not** appear in the main routing table? From which AS do these originate?
- For a BGP route that fails to appear in the routing table, what exactly does Flags say?

## Task 6: Correct multi-AS behavior

From NET3008, you will hopefully recall BGP's next-hop IGP requirement: the hand-off-point for every candidate BGP route *must* be resolvable via the IGP. Sometimes that means modifying received BGP routes before they are passed along to other peers, so that the relaying router sets its *own* address as the next hop (kinda logical: the relaying router *is* a new next hop!)

- Step 1. On R3-R4, fix the BGP next-hop resolution by means of the "next-hop-self" option. In our scenario, it only needs to be enabled on the iBGP group.
- Step 2. Don't forget to wait! Everything is based on timers, so it may take a minute or so for the network to converge.
- Step 3. Re-do the last step of the previous task. Compare to verify that the Flags field in BGP route details is cleared.
- Step 4. Then check the routing tables of **all routers**, R1-R6. Are all 11 routes available on all routers?
- Step 5. Can R1-R4 ping all subnets in their routing tables? Does traceroute help?

Questions to answer:

- Are all 11 routes available on all routers?
- Can R1-R4 ping all subnets in their routing tables?

## Task 7: Complete the route redistribution

A little more work needs to be done before route redistribution is complete. You'll need to determine the router(s) where the policy statement is required, create a suitable statement, apply it, and then make sure it is effective within the particular protocol it is used.

- Step 1. Create and apply a suitable policy-statement (on exactly two routers) so that all routes are populated in all routers in each OSPF pod. Is a single "entry" statement sufficient or do you need more than one? Remember, you're not done until you have all **11** routes!
- Step 2. Don't forget to wait! Everything is based on timers, so it may take a minute or so for the network to converge.
- Step 3. If you've waited long enough without success, you've applied a policy-option to a router. Think carefully about what *kind* of router it needs to be! Hint: review last week's postlab exercise.
- Step 4. You'll know your work is complete when you can ping from end to end (eg. R5-R6).

Questions to answer:

- How many entries are required in your policy statement?
- What *kind* of router is required for the policy to be effective?

## Task 8: Save all your configs

As always, **save your final configs** so that you can refer to them later for practice and study.

**save:** admin save ftp://{mySRLabAlphaLogin}:{assignedPswd}@mysrlab/STD/Rx/L3.cfg  
**verify:** file dir ftp://{mySRLabAlphaLogin}:{assignedPswd}@mysrlab/STD/Rx/

---