Lab 4: OSPFv2

Or: Dynamic protocols exist because configuring everything manually sucks

What You Will Do:

Working in pairs, combine your skills & knowledge to:

1. Complete basic configuration, including loopbacks and vty access for telnet + SSH
2. Configure simple single-area OSPF for IPv4 (using the interface configuration method).
3. Confirm successful OSPF neighbouring.
4. Confirm full end-to-end network reachability.
5. Successfully book and use a NetLab session to review and practice this lab's OSPF configuration.

Things that you will need to know or learn:

1. Essential device configuration – including basic security, SSH, vty access.
2. The 2 key commands for implementing OSPF, using the interface configuration method.
3. How to transfer configs between different environments: physical lab, remote access, PT.
4. Troubleshooting skills when inevitable hiccups happen with the Layer 0 and Layer 1 setup.

What you need to submit and when:

1. Complete the pre-lab quiz on BrightSpace, **before** the start of your lab period.
2. Complete the in-lab part of the exercise (see below), **before** the end of your lab period.
3. Complete post-lab exercise and quiz on BrightSpace, **before** your next lab.

Required Equipment:

- A laptop and/or a USB memory stick to save results for post-lab questions
- **Hard-cover lab notebook**, for reference during **SBA** at the end of the course.

In-Lab Marks:

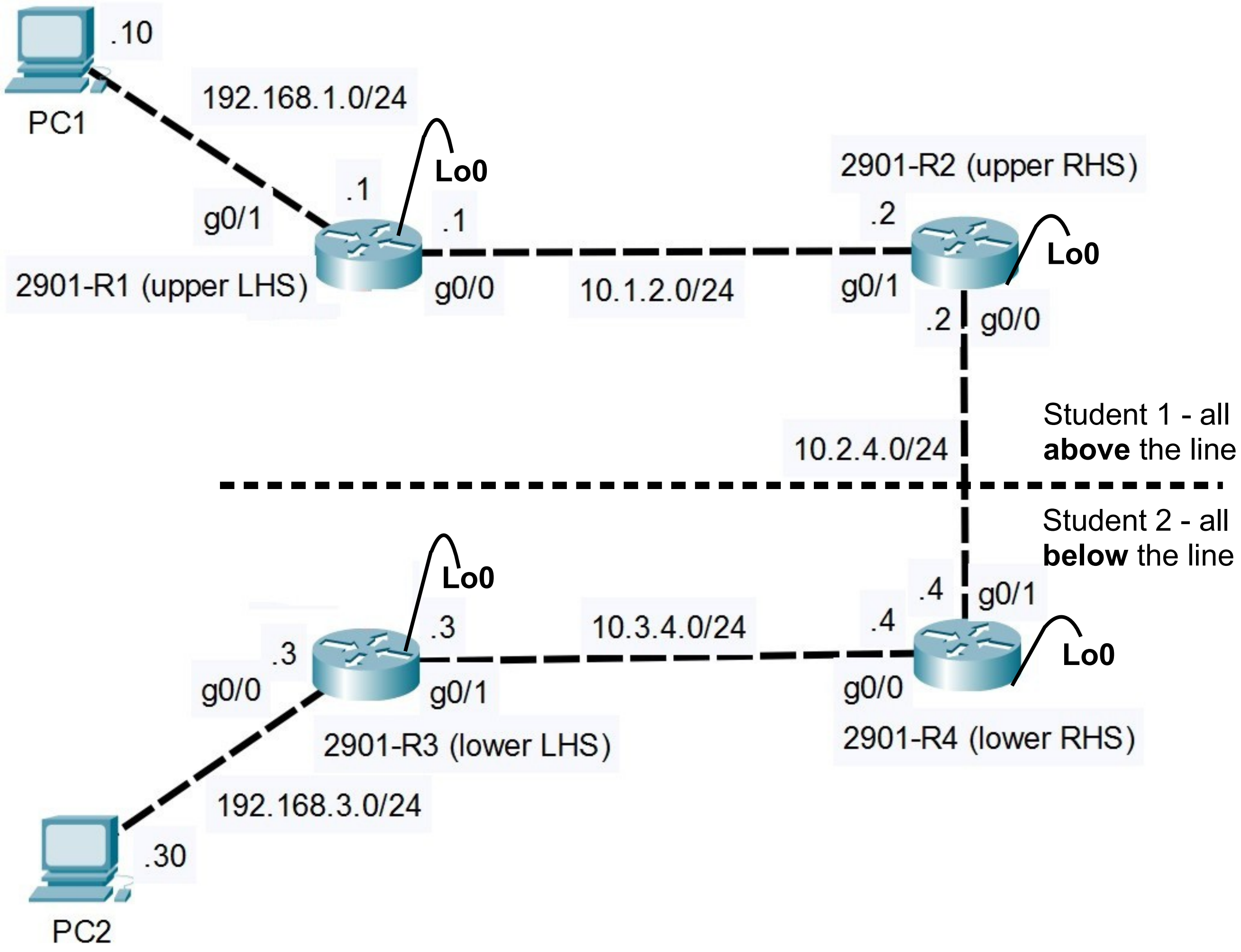
[1 mark] OSPF neighbouring, with suitable "show" command (database info) to provide proof.
[1 mark] Full end-to-end connectivity throughout your entire network.

The pre-lab is worth **35%**, the in-lab is worth **30%**, and the post-lab is worth **35%** of this lab, even though the number of points may differ between two parts.

References and Resources:

- All the materials used in your previous networking courses, including https://netacad.com
- Packet Tracer ver 8 (available from NetAcad)
- Algonquin's NetLab facility – http://netlab.algonquincollege.com

Topology and Addressing Diagram



Device	Interface	Address	Gateway
R1 (student 1) LHS rack (odd # pods): R1, or R3, or R5	Gi0/0, G0/1	As per topology	(None needed since OSPF will distribute info of all subnets!)
	S0/0/0	10.1.3.1/24	
	Lo0 – Loopback 0	**10.10.10.1/32**	
R2 (student 1) RHS rack (even #pods): R1, or R3, or R5	Gi0/0, G0/1	As per topology	(None needed since OSPF will distribute info of all subnets!)
	Lo0 – Loopback 0	**10.10.10.2/32**	
PC1 (student 1)	Realtek NIC	192.168.1.10/24	?
R3 (student 2) LHS rack (odd # pods): R2, or R4, or R6)	Gi0/0, G0/1	As per topology	(None needed since OSPF will distribute info of all subnets!)
	S0/0/0	10.1.3.3/24	
	Lo0 – Loopback 0	**10.10.10.3/32**	
R4 (student 2) RHS rack (even #pods): R2, or R4, or R6)	Gi0/0, G0/1	As per topology	(None needed since OSPF will distribute info of all subnets!)
	Lo0 – Loopback 0	**10.10.10.4/32**	
PC2 (student 2)	Realtek NIC	192.168.3.30/24	?

Task 1: Cable the Network

Cable the devices as shown in the diagram above, to create the required network topology.

Task 2: Configure basic settings for PCs

Working in parallel with your partner, determine the correct gateway address for each PC. Configure the necessary IP settings: address, mask, and gateway.

Task 3: Configure your two Routers with the essentials

Working in parallel with your partner, your two routers need to be configured with the four essential categories of basic config. This is a repeat from previous labs, Lab 2 - Postlab in particular. Please see the last page for details if necessary. The expectation is that you use copy & paste from a Notepad file, so that this takes at most 1-2 minutes.

Remember to verify connectivity between your PC and associated router. A successful ping from the PC to the router's *loopback* is rock-solid confirmation of full connectivity between PC + router.

Task 4: Configure simple, single-area OSPF

There's really only a couple of commands, repeated for each router, to *configure* single-area OSPF on a router. More interesting though is using show commands to compare before / after and see the results! READ the instructions carefully for each router (left side, right side) so that you see the differences!

Step 1. On each router, show what (dyamic) protocols are running: (**do**) **show ip protocols**
Record what, or how much, is output by the command at this stage.

Step 2. On R1 (left side) **only**: enter the OSPF process config context: **router ospf 10**
and then exit the context.
Repeat step 1 ("show ...") for *both* routers and **record** the results.

Step 3. On each router, enter the interface configuration context for the connecting interfaces (G0/0 and G0/1) and configure it as participating in OSPF: **ip ospf 10 area 0** and *quickly* repeat step 1("show ..."); **record** the results.

Step 4. After at least 30 secs has passed, note any console messages that appeared.

Step 5. Repeat step 1 ("show ...") and **record** the results.

Step 6. Examine all the output and generate some conclusions: two similar, but not identical configuration sequences were completed.
Q. Was the end result the same?
Q. Did your lab partner get the same results? If not, what is different and why?

Step 7. Test it! Ping from the PC to the loopback on R1 (or R3).
If your partner is having difficulties, help them to troubleshoot the problem.

Step 8. You should be able to telnet or SSH to both your routers. Confirm that this works and troubleshoot if not.

Q. What is the absolute, bare-bones, minimum number of different commands that are needed to get OSPF running on a Cisco router? (Admittedly, this has some limitations!)

Task 5: Confirm your understanding of OSPF

If you've stuck to these instructions, there's one piece missing for *everything* to be reachable. You'll notice the PC's *cannot* ping the loopback the far, right-side router (R2, R4). Let's fix that!

Step 1. Observe whether pings from the PC can reach the *right-side* router loopback.
Step 2. On each router, display the routing table: **show ip route [ospf]**
Examine each routing table for the route to the right-side router loopback.
Does one router have info that it's *not* sharing? Why or why not?
Step 3. We've needed so few lines of config that it's short to think about: what have we asked (configured) to be shared? Add some extra line(s) so that additional info is shared!
Hint: extra line(s) need to be repeated in a related but different CLI context.

... Is it necessary to repeat that you should **record** all your results??

Task 6: Confirm OSPF operation

Verifying OSPF (e.g. for troubleshooting purposes) can be done by showing three aspects: the participating interfaces, valid neighbours, and LSAs appearing in the OSPF topology database.

For demo purposes, you'll want to use telnet or SSH connection so you can create separate windows for demo and ongoing work purposes.

Step 1. On each router, verify the status of G0/x in OSPF: **show ip ospf interface g0/{0/1}**
Compare the results of your two routers.
Step 2. On each router, prove that you have an OSPF neighbour: **show ip ospf neighbor**
Compare the results of your two routers.
Step 3. On each router, prove that OSPF is sharing topology info: **show ip ospf database**
Step 4. On each router, prove that OSPF is working: **show ip route [ospf]**
Once you have this info, *stop* using it (it will be for demo purposes) and open a new window to continue working!

CHECK POINT: Have your window ready with the routing table, and show it to the Prof.

Task 7: Extend and verify full end-end connectivity

Find out if practice is making you faster: extend the OSPF area so that your routers and your partner's routers are all in the same area.

Step 1. Configure OSPF on your right-side router to neighbour with your partner's router.
How can you confirm the routers are successfully neighbouring? There's several different ways to prove this!!
Step 2. Display and record: the database, and the neighbour table for the right-side router.
Step 3. Verify (and troubleshoot) the connectivity: ping between PCs, and the far-end router.

CHECK POINT: Have a CMD window showing end-to-end ping between (A) the pair of PCs, and
(B) your PC and the loopback of your partner's left-side router.
(Only one demo is needed per pair of students.)

Task 8: Clearing the equipment

Be sure your equipment is cleared of all config before de-cabling and powering down.

Essential Config, VTY access, and Security

In Post-Lab 2, you created a text file with standard setup commands. That's the file you'll need to save time: copy & paste it into each router. The same USB stick that allows you to copy & paste your configs at the beginning of the lab also serves to save your configs at the end of the lab!

The required commands are:

Step 1. **Fundamentals**: (a) hostname; (b) disable DNS lookup to prevent (long, slow) lookups from incorrect commands; (c) set console logging synchronous; (d) MOTD clearly restricting access to authorized users.

Step 2. **Security**: (a) Exec ~~password~~ secret of cisco; (b) ensure plaintext passwords are all encrypted automatically; (c) console password of cisco; and (d) console exec-timeout 120.

Step 3. **IP addressing**: (a) loopback interface Lo 0 and/or 1 (other interfaces will vary)

Step 4. **VTY access** (to eliminate shuffling the console cable in lab).
SSH access requires 5 additional commands:
(–) hostname [already done in step 1, so here only for documentation purposes];
(a) domain name [e.g. NET2000.com];
(b) RSA key with max allowed key length (CLI cmd= **`crypto key generate rsa ...`**);
(c) a user account with user/pass of **cisco** / **class** (CLI cmd = **`username ...`**);
(d) **`login local`** on all VTY lines;
(e) **`transport input ssh`** on all VTY lines

Step 5. **VTY fundamentals and security**: (a) set logging synchronous; (b) set an idle timeout of 2 hours.

Step 6. **VTY access** via Telnet (for backup) requires only 3 commands: (a) defining an account [done]; (b) **`login local`** on all VTY lines; (c) allowing Telnet (possibly in addition to SSH: **`transport input ssh [telnet]`** (N.B. Real routers allow you to stack multiple options in a single command, but PT does not. Use CLI help to find the way to allow both SSH and Telnet in PT.)

Step 7. Test telnet: from a command window on a PC or the CLI on another router, use the **`telnet x.x.x.x`** command to reach the router you've just configured. For troubleshooting, you *know* you can reach the router (right?!!) so you just need to check the config specific to telnet. Verify that you can enter Privileged Exec mode.

Step 8. Test SSH: from the CLI on another router, use the command **`ssh -l {user} x.x.x.x`** to reach the router you've just configured. Don't forget that the {user} was configured in step 4.

Step 9. When connected via SSH or telnet, be sure to set **`terminal monitor`** in Privileged Exec mode so that you see log messages! Test it by creating or shutting down an interface (e.g. Lo2); if you don't see the up/down messages for that interface then it's not working. Troubleshoot.