Lab 2 Postlab: Default Routes

or: *Getting "extra time" for lab work*

What You Will Do:

1. Replicate the full topology from Lab 2, with the addition of loopback interfaces
2. Complete the configuration for basic connectivity for a ring of 4 routers
3. Observe the differences in output between PCs and Cisco devices when running ping and traceroute.
4. Build upon your knowledge and understanding of ping and traceroute.

Things that you will need to know or learn:

1. Essential device configuration – hostname, MOTD, Loopback/Vlan1, *basic security* , IPv4
2. The convenience of loopback interfaces for network troubleshooting
3. Designing and implementing default static routes for small or stub networks
4. Understand how ping and traceroute work, key differences, and what they can tell you.

What you need to submit and when:

1. Complete the “Lab 2 Post-lab” quiz on Blackboard with answers from your work, **before** your next lab.

Required Equipment:

- Access to a computer running a recent version of Packet Tracer (e.g. ver 8.2).

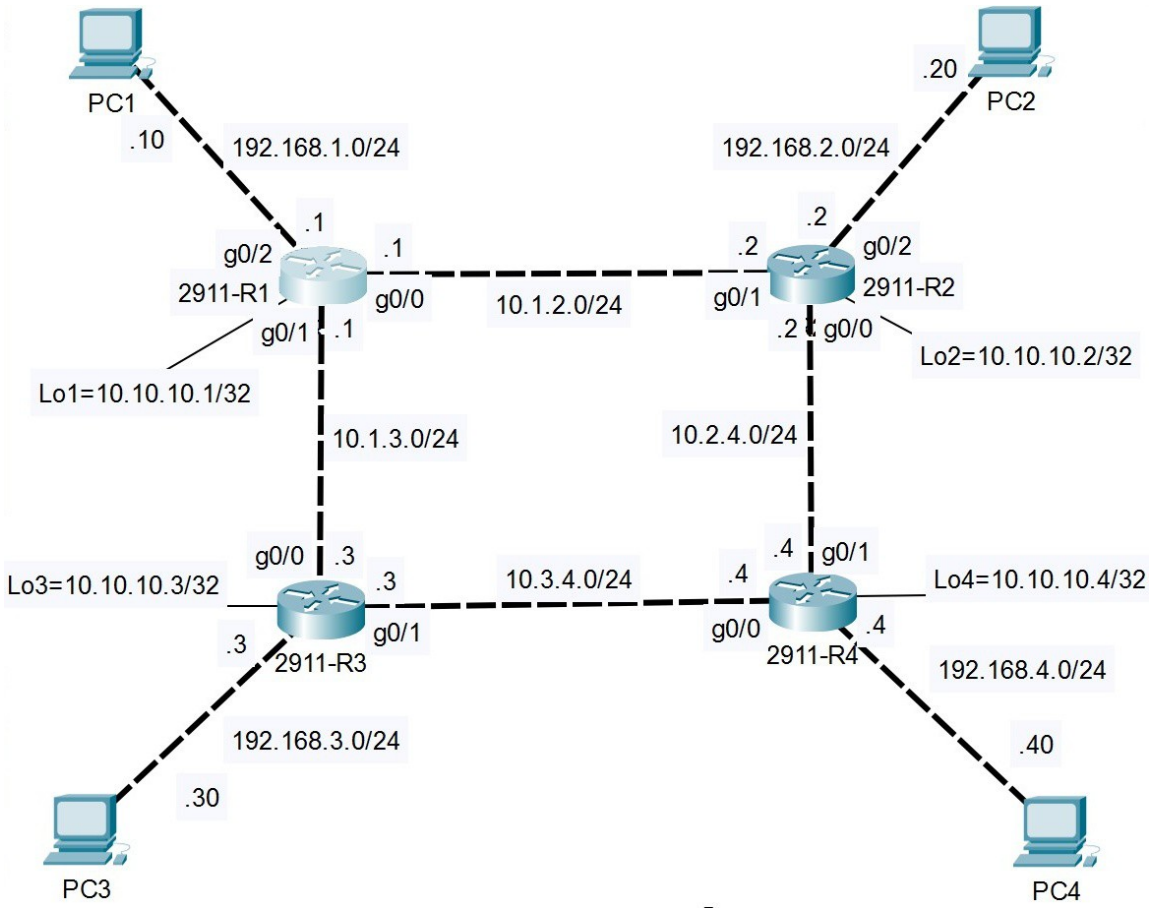
Marks:

This post lab is worth 35% of the mark for Lab 2.
10% of your final mark is for labs done during the course of the semester.

References and Resources:

- a recent version of Packet Tracer (e.g. ver 8.2 or higher; available from NetAcad.com)
- NET1006 notes on default gateway and default static routes

Lab 2 Postlab – Topology and Addressing Diagram



Device	Interface	Address	Gateway
R1 (**2911**)	Gi0/0	10.1.2.1/24	default route goes to **next router clockwise** address = **??**
	Gi0/1	10.1.3.1/24	
	G0/2	192.168.1.1/24	
R2 (**2911**)	Gi0/0	10.2.4.2/24	default route goes to **next router clockwise** address = **??**
	Gi0/1	10.1.2.2/24	
	G0/2	192.168.2.2/24	
PC1	FastEthernet	192.168.1.10/24	?
R3 (**2911**)	Gi0/0	10.1.3.3/24	default route goes to **next router clockwise** address = **??**
	Gi0/1	10.3.4.3/24	
	G0/2	192.168.3.3/24	
R4 (**2911**))	Gi0/0	10.3.4.4/24	default route goes to **next router clockwise** address = **??**
	Gi0/1	10.2.4.4/24	
	G0/2	192.168.4.4/24	
PC3	FastEthernet	192.168.3.30/24	?

Task 1: (Re-)Create the Topology in Packet Tracer

Make sure you have a recent version of PT. If need be, you can download it using your account on NetAcad.com. Be sure to use **2911** model routers since they have **3 ethernet** interfaces.

Objective: Though you may see creating the topology as unnecessary or tedious work, it's exercising the mental muscles involved with:

- digging through all the details you'll typically see in a network diagram;
- being *accurate* when cabling devices (is that G0/0 or G0/1 I need to use for this cable?);
- being accurate when configuring devices that are very, very similar but not completely identical!

Don't forget to add the **loopback** interfaces on each router.

... And we will likely use this same topology for working with IPv6, so you'll get a double payback for your time and effort.

Task 2: Configuring the PCs

The PCs only need their IPv4 address, mask, and default gateway configured. You'll need to determine the default gateway address (same as for Lab 2).

Task 3: Configuring the Routers

Like you did for Lab 2, you'll need to configure the **addressing** for four interfaces for each router (G0/0, G0/1, G02/, loopback). You'll also need to set a **default static route** on each router.

For convenience, you may want to set the console connect to *never time out*, so that you can instantly return to the CLI for any device.

Task 4: Connectivity between each PC and it's router

The best way to verify, and start debugging if necessary, is to confirm connectivity between each PC and it's associated router.

Step 1. Open a CMD window on the PC and ping the router's G0/2 interface (i.e. the default gateway for the PC). If it doesn't work, the work you'll need to verify is confined to a small area: IP address/mask for the PC; and the correct interface (G0/2) & address/mask on the router. Are all interfaces *no shut*? Continue troubleshooting until all PCs can ping their gateway.

Step 2. Ping the router's loopback interface. If needed, check the loopback address/mask. Continue troubleshooting until all PCs can ping their loopback.

Task 5: Connectivity between adjacent routers

From the router CLI, confirm connectivity from each router to it's immediate neighbours. Ping both the address for the directly connected interface, as well as the loopback. Troubleshoot by checking that cabling goes to the correct interfaces, address/mask are assigned to the correct interface, and that all interfaces are up/up.

Task 6: Full Connectivity Verification

All the pings so far tested connectivity between adjacent devices (i.e. both on the same wire). Next we want to test end-to-end connectivity. For teaching / learning purposes, the routing we're using in the ring topology is set up in a specific way. That creates an *all or nothing* situation: either it all works, or none of it works (though this is **not** the case with all ring topologies!)

The (four) default static routes are the only thing that makes or breaks connectivity at this point!

Step 1. Have another look at the default static route that you've configured on each router. Make sure there's no typos or other mistakes.

Step 2. Try pinging from any router to the diagonally opposite router. Try both the loopback and a physical interface address. If it works, then Congratulations! If not, and you've completed all the previous tasks, you'll have to go back to step 1 for troubleshooting.

Step 3. Do a complete verification of connectivity by filling out the charts below. Enter "NoRoute" for any tests that fail. **Note** carefully the order of the routers and PC. (Wow! Those tests are pretty repetitive! Maybe that's something you'd like to automate using knowledge gained in NET2008?!!)

Step 4. See if you can determine the pattern(s) in the results. Once you notice the patterns, can you figure out the reason *why* for each result?

[And by the way, in recognition of the skills you've demonstrated, next time the instructions for Task 1-6 will be: "Configure the network and troubleshoot until it's working correctly." :-)]

Tests run from PC 1

Test type	R1	R2	**R4**	R3	PC3
Traceroute (total number of hops)					
Ping (TTL value reported, if any)					

Tests run from PC 3

Test type	R3	R1	R2	R4	**PC4**
Traceroute (total number of hops)					
Ping (TTL value reported, if any)					

Tests run from R1

Test type	R1	R2	**R4**	R3	**PC3**
Traceroute (total number of hops)					
Ping (TTL value reported, if any)					

Task 7: Standard Security & Config: Prepare for future labs in T108

Everything is working perfectly, so get some added benefit from your work. You'll save a ton of time if you have a copy of all required commands in a text file for ease of copying & pasting!! These commands will be used *every week, on every device* from now on. For each command below, enter the full, un-abbreviated command in the CLI and then when it works correctly, copy that exact command into a text file (e.g. notepad).

All of the required commands are hopefully review from previous courses.

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 Lo1 (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 (CLI cmd = **`user ...`**);
(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.

Now be sure that you've copied all those commands into a text file and save the file on a USB stick!!