

Lab 3: IPv6 – Addressing, DHCP, Static Routes

or. Same functionality but with longer addresses than IPv4

What You Will Do:

1. A. Enable, configure, and identify IPv6 link-local addresses on both a router and PC
B. Ping a link-local target address from both a router and a PC
2. Configure and examine static, SLAAC, stateless DHCP, and stateful DHCP addressing
3. Observe which address (link-local, global unicast) a DHCPv6 client uses for the GW, DNS
4. (Stretch goal) Configure IPv6 static and default routes

Things that you will need to know or learn:

1. How to turn off the firewall in MS-Win.
2. How to access the Properties window for a NIC under MS-Win 10.
3. How to copy & paste inside a CMD window on MS-Win.
4. A clear understanding of the difference between link-local vs Global unicast addresses.
5. Meaning of the written format for IPv6 addresses and masks.
6. Essential IPv6 address ranges:
Link-local (FE80::/10), Global unicast (e.g. 2001::/64), Site local (FD00::/8), Default (::/0)
7. There's no pre-defined interface (or location) for link-local addresses!
8. Wading through the (longer) output of commands when using IPv6.

What you need to submit and when:

1. Complete the Lab 3 pre-lab quiz on BrightSpace, **before** your lab period.
2. Complete the in-lab checkpoints (see below), **before** the end of your lab period.
3. Complete the Lab 3 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:

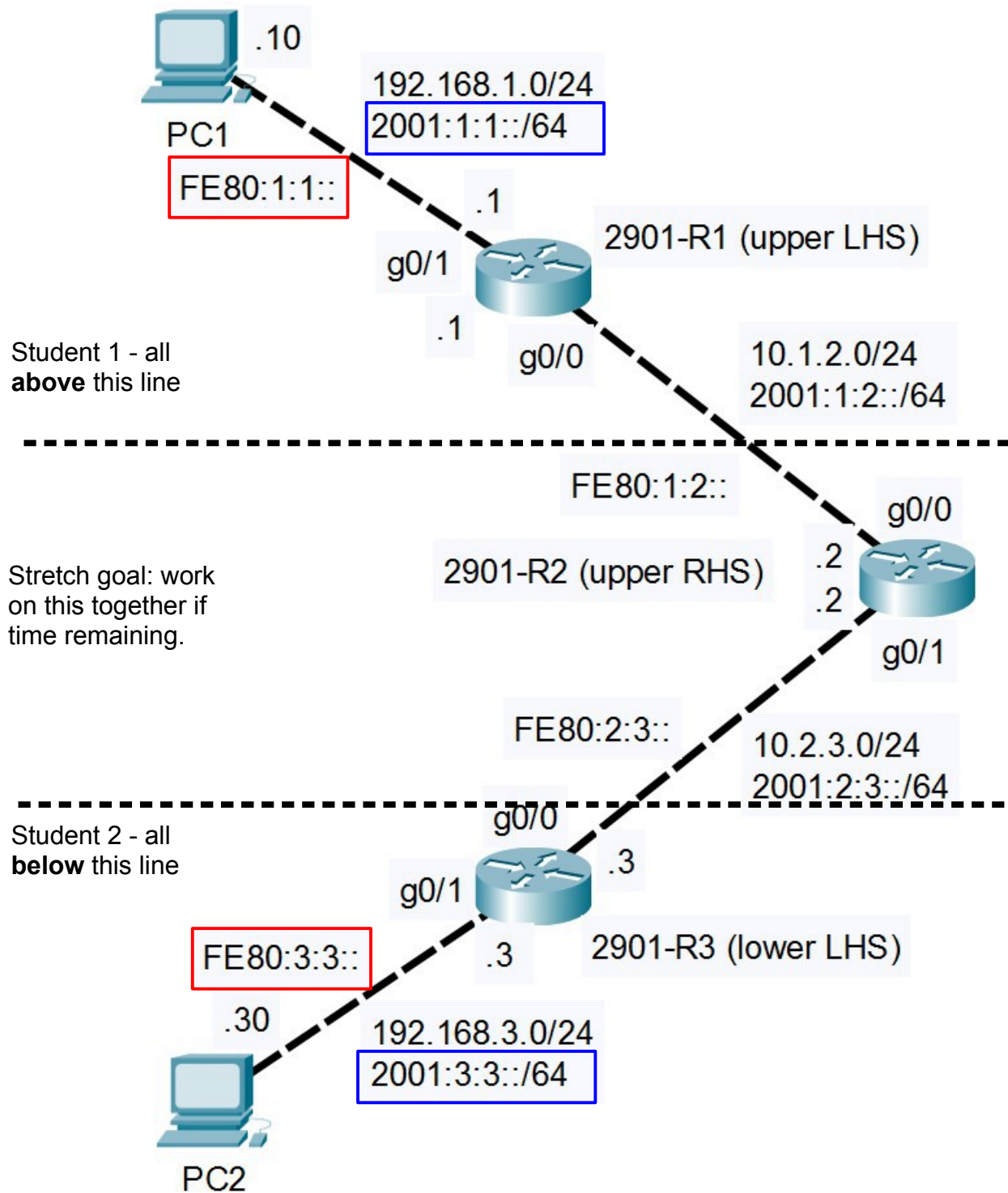
- [1 mark] Demo successful IPv6 pings to a link-local address, sourced by the PC **and** the router.
- [1 mark] Demo successful SLAAC addressing on the PC.
- [1 mark] Demo successful stateless DHCPv6 on the PC.
- [1 mark; Bonus] End-to-end ping from PC to PC via R2

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

References and Resources:

- Wk02 slide deck on IPv6
- Materials used in your previous networking courses, including <https://netacad.com>
- Packet Tracer ver 8.2 or higher (available from NetAcad)
- Algonquin's NetLab facility – <http://netlab.algonquincollege.com>

Lab 3 – Topology and Addressing Diagram



Device	Interface	Address	Gateway
All routers	Gi0/0, G0/1	As per diagram above	R1, R3 = GW for PC R1, R3 – default routes R2: uses static routes only.
	Loopback 0 (not drawn)	10.10.10.{R}/32	
	Loopback 1 (not drawn)	FD00:10:10:10::{R}/128	

Task 1: Cabling, Basic Configuration, Verification

Working in pairs (entire lab), sign up for your pod and set up the topology from the previous page:

- Cabling
- Basic configuration (as per Lab02-PostLab): Fundamentals, Security, VTY access (Seriously, just copy & paste the configs you've made, with two edits in each of R2, R3 for the changed interface and addressing!)
- Although IPv4 addressing is not required, it will give you confidence and ensure that everything is set up properly. You can move to the next step when each PC can ping (via IPv4) it's associated router, and each router can ping it's adjacent neighbor(s).

Task 2: IPv6 under MS-Win 10

In this lab, the PCs play the role of DHCPv6 clients. But even before that happens, the PCs will generate their own link-local IPv6 address which can be observed and ping'ed. Work in parallel with your partner, each person on their own PC.

Step 1. Use your favourite method for accessing the Properties window for the Realtek NIC (a series of clicks starting from the Networking icon on the task bar, or via the Settings item appearing in the Start Menu fly-up)

Step 2. Ensure the IPv6 entry has a check-mark which means that the IPv6 protocol stack is enabled and active:



Step 3. Open a CMD window, and display the complete IP settings: `ipconfig /all`
With our lab setup, there will be a lot of output. We only need the first section ("Windows IP Configuration") and the section for the Realtek NIC (it starts with "Ethernet adapter", so ignore anything starting with "Wireless ..." or "Tunnel adapter"). **Copy and save the output** for comparison later in this lab, and for possible post-lab questions.

Step 4. Find and make note of two addresses in particular:
– the NIC's MAC address listed as "Physical Address"; and
– the **link-local address**, listed as "Link-local IPv6 Address"

Step 5. Take a few moments to understand the output that appears:

Q: What special / unexpected info is included in the link-local info? What does it mean?

Q: On MS-Win, is the low-order half of the link-local address based on the MAC address or is it randomly generated (e.g. for privacy)?

Task 3: Enabling IPv6 on a Cisco router

You may find it surprising, but Cisco routers (and switches!) are **not** enabled for IPv6 by default. IPv6 on Cisco's routers is simpler than on their switches: IPv6 is enabled on a per-interface basis, and actual routing capability is activated system-wide by a separate command.

IPv6 is enabled on an interface by one of two methods: (a) the command `ipv6 enable` or (b) by setting a unicast IPv6 address (either link-local, global unicast, or site local).

Step 1. Enable IPv6 on the G0/1 interface by setting a link-local address:

```
ipv6 address FE80:R:R::R link-local
```

where: **R** is the router number (1, 3) as per the topology diagram.

Step 2. Verify (ALWAYS!) the setting and value is correct, and fix mistakes as necessary:

```
show ipv6 interface brief
```

Task 4: Confirming Connectivity with Link-local addresses

Ping is often a great way to confirm connectivity. We'll use the same command you're already familiar with, and just use an IPv6 address instead of IPv4.

When using link-local addresses on Cisco devices, you'll be prompted to provide the **full** unabbreviated interface name to be used for the ping request. No spaces or spelling mistakes, are allowed and sometimes the i/f name is case sensitive (on real equipment, but not on PT).

You need to know two things about MS-Win:

- The "%XX" that you saw on the NIC's link-local address identifies the interface number. Use that as a **suffix** for the target's link-local address (i.e. router's address).
- Be WARNED that by default MS-Win does **not** respond to pings: the firewall must be disabled (or the rules modified) in order to allow ping requests and thus generate ping replies!

Step 1. Open a CMD window on the PC and issue a ping command:

```
ping FE80:R:R::R%xx (Watch those single and double colons!!!)
```

Troubleshoot if need by by verifying correct cabling, interface addresses; no typos!

Copy and save the output for comparison purposes.

Step 2. From the router CLI, issue a ping to the PC's link-local address (Task 2, step 4)

```
ping FE80:x:x:x:x:x:x: (See last page for some tiny/crazy details)
```

The PC is connected to G0/1, so use that when asked about which interface to use. Troubleshooting should focus on typos, and confirming the MS-Win firewall is off.

Copy and save the output for comparison purposes.

Step 3. Compare the information available from the two devices (MS-Win, IOS).

Q: What is common to both and what appears in only one of the outputs?

CHECK POINT: With the output **already on your screen**, show the ping results to the lab Prof.

Task 5: SLAAC

Link-local addressing stays on a single link and doesn't work for multi-hop destinations. For that globally unique addressing (or even site local) is required. In this lab, we'll start with SLAAC first and then move along to a client-server DHCPv6 setup. SLAAC setup is very easy for both the PC and router.

Step 1. Confirm / set the PC to respond to RA messages. In Task 2, step 2 we confirmed that IPv6 was activated. Return to that window and this time dig a little deeper by double-clicking the IPv6 item. The default for MS-Win is that both the address and DNS server are obtained automatically. Ensure they are set to automatic (if they were changed from the default.)

Step 2. Set a routable address/mask and enable RA messages on the router.

A. Set the IP address and mask using a modify version of the "ip address ..." command you know well: `ipv6 address 2001:R:R::R/64` (where R = your router number)

B. Issue the command `ipv6 unicast-routing` to make the device a "real" router: (1) capable of routing, and (2) also capable of generating RA messages.

Step 3. Allow some moments (for internal timers to complete their count), then check that the PC has adopted an address within the router's subnet (based on the RA message).

Repeat Task 2 step 3 and view PC addresses: `ipconfig /all`

Look for the lines starting with "Autoconfiguration Enabled" and "IPv6 Address"

Step 4. If you've waited 1-2 mins with no luck, try giving MS-Win a kick: try disconnecting and reconnecting the cable for the Realtek NIC, or doing "ipconfig /release" and "ipconfig /renew". Remember to pause for ~5 secs between each action. As the ultimate action, you can reboot the PC (**after** you've saved any results from previous steps!)

Step 5. Assuming you've obtained an address in the expected range, **copy and save** the output of the "ipconfig /all" command. Congratulations: you have SLAAC working!

Q: Does the PC use the global unicast or the link-local address for the gateway?

CHECK POINT: With the output **already on your screen**, show proof of SLAAC to the lab Prof.

Task 6: Stateless DHCPv6

Configuring stateless DHCP only requires a few lines of config on the router. Maybe the biggest challenge will be getting the MS-Win to cooperate with refreshing the addressing! Note that DHCPv6 is configured on a per-interface basis.

Step 1. The following 6 lines configure the DNS address and domain name:

```
Rx(config)# ipv6 dhcp pool STATELESS-SETUP
Rx(config-dhcp)# dns-server 2001:R:R::R (e.g. your router's G0/1 interface)
Rx(config-dhcp)# domain-name STATELESS.com
```

```
Rx(config)# interface g0/1
Rx(config-if)# ipv6 nd other-config-flag
Rx(config-if)# ipv6 dhcp server STATELESS-SETUP
```

Step 2. Once that's done, we'll need to poke MS-Win to get it to refresh.

Try all the steps in step 3-4 of the previous task.

Step 3. Once the change of settings on the PC is visible (DNS server IP and domain name), **copy and save** the output of "ipconfig /all"

Q (later?): What command above sets the "O" bit? Ref: Wk02Day2 lecture summary notes

Q: Does the PC use the global unicast or the link-local address for the gateway and DNS?

CHECK POINT: With the output **already on your screen**, show the stateless DHCP.

Task 7: Stateful DHCPv6

Stateful DHCP configuration is quite similar to stateless DHCP. Compare the two configurations!

Step 1. There's 6 actual lines to configure stateful DHCP:

```
Rx(config)# ipv6 dhcp pool STATEFUL-SETUP
Rx(config-dhcp)# address prefix 2001:R:R::/64
Rx(config-dhcp)# dns-server 2001:R:R::R (e.g. your router's G0/1 interface)
Rx(config-dhcp)# domain-name STATEFUL.com
```

```
Rx(config)# interface g0/1
Rx(config-if)# no ipv6 nd other-config-flag
Rx(config-if)# no ipv6 dhcp server STATELESS-SETUP
Rx(config-if)# ipv6 nd managed-config-flag
Rx(config-if)# ipv6 dhcp server STATEFUL-SETUP
```

Step 2. As before, refresh the PC's addressing, then copy & save the output.

Q: What command above sets the "M" bit Ref: Wk02Day2 lecture summary notes.

Q: Does the PC use the global unicast or the link-local address for the gateway and DNS?

Task 8: (Optional; Stretch Goal)

If you still have at least 15 mins, can you:

- Assign the designated IPv6 addresses on R1, R3 G0/0 interfaces
- Assign the designated IPv6 addresses on both interfaces on R2
- Create a default route on R1, R3 pointing at the respective interface on R2
- Create 2 static routes on R2: one for each of the IPv6 subnets on R1, R3 G0/1 interfaces
- Prove it all works by pinging from end-to-end i.e. PC to PC.

CHECK POINT [Bonus]: With the output **already on your screen**, show the ping to the lab Prof.

Task 9: Finishing up and Clearing the equipment

1. Make a backup of your configs; it'll likely come in handy for the postlab!
2. BE SURE THE EQUIPMENT IS CLEARED of all config before you de-cable and power off

Homework:

- Make a list of commands for **configuring** IPv6 addressing and DHCPv6 setup, and a list of **show** commands for **displaying** these same settings (on either the PC or the router). You'll need it for the SBA!!
- **Q:** Check your understanding: It was *never* necessary to explicitly configure a default gateway address in all three of the addressing setups (SLAAC, stateless, stateful). In IPv4, it would have *always* been necessary. Why didn't we ever need to configure the gateway in IPv6? Is it magic or something?

Pinging link-local addresses from CLI

```
Router(config-if)#do ping FE80::C664:13FF:FE68:55F1
```

```
Output Interface: gigabitethernet0/1
```

Lowercase works, when prompted

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to FE80::C664:13FF:FE68:55F1, timeout is 2 seconds:
```

```
Packet sent with a source address of FE80::C664:13FF:FE68:55F1%GigabitEthernet0/1
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

```
Router(config-if)#do ping FE80::C664:13FF:FE68:55F1
```

```
Output Interface: GigabitEthernet0/1
```

Uppercase works, when prompted

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to FE80::C664:13FF:FE68:55F1, timeout is 2 seconds:
```

```
Packet sent with a source address of FE80::C664:13FF:FE68:55F1%GigabitEthernet0/1
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

```
Router(config-if)#do ping FE80::C664:13FF:FE68:55F1%GigabitEthernet0/1
```

Uppercase when all-in-one

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to FE80::C664:13FF:FE68:55F1, timeout is 2 seconds:
```

```
Packet sent with a source address of FE80::C664:13FF:FE68:55F1%GigabitEthernet0/1
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

```
Router(config-if)#do ping FE80::C664:13FF:FE68:55F1%gigabitethernet0/1
```

NO lowercase when all-in-one

```
% Unrecognized host or address, or protocol not running.
```