Copyright Notice:  
All images and screenshots used in this document are the property of CBT Nuggets. Used for educational purposes only. This document is licensed under a Creative Commons Attribution 4.0 International License.

**Part 1:** A new computer has been set up on network 10.10.0.0/24. A screenshot from one of the PCs set up on the network shows that the default gateway/router doesn’t have an IP address- as ‘Destination Host Unreachable’ means that the packet arrived at its location, but was dropped due to wrong address- a ICMP ping was sent back as a result ‘Lost’:

A computer screen with white text

AI-generated content may be incorrect.

To resolve this issue, I first entered the router’s terminal configuration mode from the Server in MTPuTTY to view each of router 1’s interfaces and their assigned IPs/Subnet masks/connection types:

A computer screen shot of a computer program

AI-generated content may be incorrect.

Comparing those interface configurations to the network map, it’s clear that default gateway address for the 10.10.0.50 client computer hasn’t been configured on the router it’s physically connected to. To fix this, I assigned an available interface on the router (0/2) to 10.10.0.1, as is customary when setting up a router on a subnet- it’s usually the first few available in the host id part (this is a class A private ipv4 network, we’ll see ipv6 examples later):

A screenshot of a computer program

AI-generated content may be incorrect.

After running ‘config t’ on this router, I first used do show ip router connected, and on it, I saw that the 10.10.0.0/24 network is available on that interface 2/0- and the cable to the switches for the computer I pinged before was plugged into that port on the router. I then selected the interface with ‘int gigi 2/0’, and set the default gateway for that network to 10.10.0.1 (first available host on the network). To verify it worked, I ran ‘do show ip route connected’ command to verify the interface was set, and you can see that 10.10.0.1 in the image above is locally connected (compare to ‘link local’ ipv6 addresses), on interface 2/0. The ‘/32’ refers to the host byte ‘.1’ of the ipv4 address for Router 1 on that interface- often called the ‘block size’. If ‘255.255.255.0’ doesn’t set the mask right, the ? can be used to figure out what syntax to use, for instance, in some vendors it could be /24.

Now, from the 10.10.0.50 client PC plugged into the router at the 2/0 interface, when I ping the default gateway of 10.10.0.1 or any other device on the network, it can tracert or ping them!: A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

Worth noting is that sometimes with networks we have to be creative- to slice & dice the subnet masks such that it fits a range of devices in certain areas/interfaces of a network- including different VLANs and network topologies (people should always remember to disable VLAN 1 if they use any kind of trunking- even a child could hack into that!

**Part 2:** Subnet a network across the interfaces in a group of routers on the network 10.67.83.0/24 (task assigned to you by administrator/engineer, likely in response to the need to add more users to each room for business expansion, or even a bunch of printers and servers):

A diagram of a network

AI-generated content may be incorrect.

Given there were counted, either by the Engineer or myself, the need for 4 new subnets across the building (5 others are already configured and in use): in-between the routers and different cabling all around: Green for gigabit, yellow for Fast Ethernet, red for Serial, and all the routers, PCs, Servers, and associated switches- all being part of the same physically connected all being part of the same 10.67.83.0/24 network (run on sentence)… I went to each router’s configuration settings, one-by-one, and configured the 4 new subnets on the available interfaces that according to the network map go to different groups of switches in the corresponding areas that divide up to the PCs/devices- all in different rooms/areas of the buildings (again, that is not always the job of a network technician, but an engineer/router group person- it depends on the company). The first interface I configured was **Gig 2/0** on **R1: A close-up of a number

AI-generated content may be incorrect.**



In this process, I ensured that I checked before and after what interfaces were up or down:

A close-up of a computer screen

AI-generated content may be incorrect.



As you can see, the 10.67.83.1 interface for the ‘default gateway’ was configured on Gig 2/0, and is ‘up’, whereas it wasn’t present before. The broadcast address of that range (e.g. for DHCP Discover Packets, among other uses) would be 10.67.83.15, as the .240 (/28) of the subnet mask for the IP indicates that the block size is 16 for each of the 16 subnets in this configuration- the last 7 which are not configured yet so spare ‘room’ for the future, or that the subnetting rules wouldn’t allow it (/27 = 8 subnets, /28 = 16 subnets).

On the client PC connected to that interface, you can see that 10.67.83.1 is now its default gateway, but it can’t be pinged by other devices on the network yet:

A screen shot of a computer

AI-generated content may be incorrect.

I then configured the new, ‘faster’ **Gigabit 2/0** interfaces on the network map at PC-20, PC-30, and the server to the proper subnets.

A screenshot of a computer

AI-generated content may be incorrect.

These are manual configurations, so they appear distinguished as ‘manual’ on the method row. Commands like show ip interface brief, among many other commands, can’t be used in (config) or (config-if) modes- and also, I type super-duper fast (88 wpm, 99% accuracy).

With the 4 interfaces configured on each router, as requested, with the 4 subnets & the default gateway in each one, it’s now possible for multiple devices to be connected to them in each subnet, up to 14 different ones. Here’s proof my configuration works now when using the ping command from PC-20 & Server in the 10.67.83.129 subnet from above: A screenshot of a computer code

AI-generated content may be incorrect.

Server:

A computer screen with white text

AI-generated content may be incorrect.

While Ipv4 private ranges of class A, B, or C are used widely still, the move to ipv6 is imminent, and it comes with both security & compatibility features.

**Part 3:** Ipv6 Network Configuration

Ipv6 Addresses are more difficult than ipv4 to read at first, which is why hands-on experience in live systems remote or virtual with good in-the-field professional training like CBT Nuggets or CompTIA is crucial- a college environment is rarely suitable, from my observations. Maybe not MIT. Still, the internet is a wonderful resource and knowing that these resources do contain better or ‘alternative’ explanations, often more hands-on, for concepts and job tasks- and that one day I’ll build and maintain the path of that information to people- and likely the sensitive or critical value, is a good way to kick off this ipv6 network configuration project!  
For future reference & full understanding, here is a chart of the different types of ipv6 addresses:

A table with text and numbers

AI-generated content may be incorrect.

In ipv6 there are, as the A+ and Network+ both said, **no broadcasts** allowed. There are, on each individual interface, a Global Unicast and Link-local address which are both routable on the Internet. Unique Local is equivalent to the ipv4 private address ranges of A, B, & C.

Before we talk about ipv6 addresses, it’s important to understand a crucial topic first: how addresses get configured/defined as ipv6. Instead of hosts/endpoints using ARP (Address resolution protocol) tables for MAC addresses that get mapped to IP addresses on the local network like ipv4, ipv6 instead uses NDP (Neighbor Discovery Protocol). The NDP protocol can look for router IP addresses with ‘router solicitation’ request packets, or host IP addresses with ‘neighbor solicitation’ request packets to obtain their MAC addresses. The host devices and routers can give ‘neighbor advertisements’ in response to these solicitations- and routers will do it on a timer- updating the MAC address in the neighbor discovery protocol table. This leads to a good point:  
*Instead of relying on the possibly insecure or manipulatable DHCP Servers*, **ipv6** with **NDP** does something to the hosts/routers called SLAAC (Stateless Address Auto Configuration). This works by using the barrage of back-and-forth neighbor solicitations and advertisements on a network to figure out unique ipv6 addresses for each device, on the proper subnets, and on the one and only default gateway for each network.

…which gets to my point, Multicast ipv6 addresses (not configured on interfaces, automatic) are, in the first 16-bits/4 nibbles, FF00::-FFFF::/8 with some exceptions- FF02::1 is from **hosts** **or any other device** configured with ipv6 addresses, and FF02::2 is from the **routers**. Additionally, FF02::5 means it’s a multicast address configured with the link-local routing protocol Open Shortest Path First (OSPF). Think of it in terms of the neighbor solicitation- there will be a hands-on lab on this in a few pages. A-F is 10-15 or 1010-1111. **The /8 bits means so long as it starts with the first 2 nibbles- FF, it’s a multicast address**. **The /64, /7 and others use the same logic.** It would be wise for beginner network technicians to always keep this table/chart with explanations available at anytime, as ipv6 networks become more and more common, despite ISP’s and other companies insistence on using private ipv4 ranges with NAT (Network Address Translation) to their site, along with cloud migration too.

To view the neighbor tables on windows (instead of arp tables with arp -a), you run these commands in the command prompt:A screenshot of a computer

AI-generated content may be incorrect.

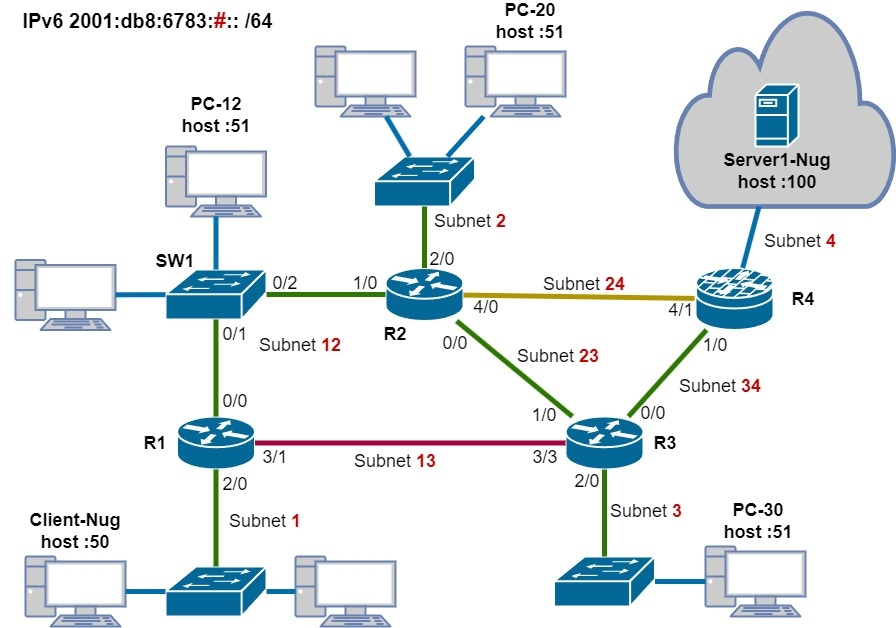
Netsh is somewhat like the different router configuration screens, just on the host level… you can effectively turn Windows computers into ‘virtual routers’, or a hotspot, with the right network adapter configurations.

Also, you check certain registry entries in Windows Machines to see some of these addresses, including ipv6 addresses derived from MAC addresses, which is what EUI-64 in SLAAC does! EUI-64 is used in SLAAC of a host if link-local addresses are hard to find without them already being in use- it’s derived from the MAC address of the particular device, taking the Hexadecimal MAC address and adding 'fffe’ in the middle of it.

Last, it’s important going forward in this next lab to know that each and every ipv6 interface has two ipv6 addresses on it: A Global unicast and a link local address (in the ranges of the table above).

**Lab:** Ipv6 Configuration

Given Network Map:



The # symbol corresponds to the subnet of the network id 2001:db8:6783 network. The 4 hexadecimal characters after the 3 sets of 0000 (::) identifies the server/pc on that subnet, but the entire host id is all of the characters following the : after #. This host can have a default gateway either configured link-local to the subnet id of that connection cable/type, or to the global unicast address of the router. It is best to configure the link-local address on the router’s interface the connection is for troubleshooting purposes (router terminal)- you can view it there, even remotely- and don’t have to travel far and get additional permissions for manually configuring the ipv6 address on the host. I put this into practice on the network shown in the map:

**Issue:** Your company has just set up a new router, and it needs configuring- as well as the devices on the interfaces connected to it.

**Solution:** I’ll configure it from the MTPuTTY program for SSH sessions/in-band management on all the routers and devices- ready to configure from a single location without having to worry about traveling to all of those PCs and devise to configure them there. Ipv4 has a Default Gateway address on each interface with a network address on each router… so in this case, we’ll set link-local addresses to each interface on the routers, and a global unicast address on them too for routing over the internet. This way, the actual local link address it is will not be known across the internet.

Steps & Screenshots:

A computer screen shot of a computer

AI-generated content may be incorrect.

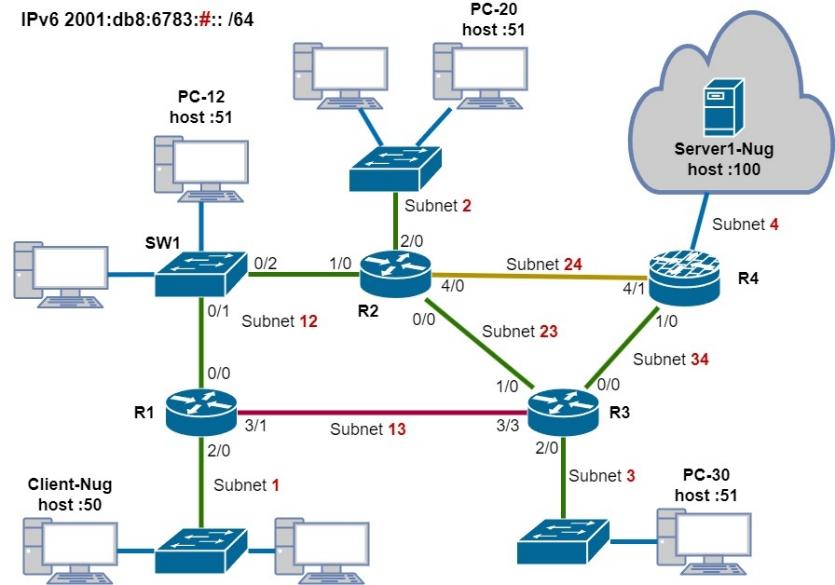
Notice how “end” can exit all the way out of configuration mode. It’s important to find shortcuts on the respective vendor brand, as building networks can take a while to finish.

After configuring each interface, I checked with PC-20 if it could ping/trace the other devices across the network by their ipv6 addresses (using the appropriate subnet & host ids):

A screenshot of a computer

AI-generated content may be incorrect.

Here’s the network map again to put things into perspective:



Notice how Router 2, there’s the global unicast ipv6 address I configured “2001:db8:6783:2::2/64” as the first hop in the trace command, not the link-local address of FE80::2. The /64 in this case refers to the last 64 bits of the ipv6 address which is the host section, the :: followed by 2 (for R2).

***In another lab, I will configure all of the Dynamic and Static routing protocols on a network*** and experiment with how they would work, and why we would use each of them given a certain requirement in designing the network for the company, hence, what these indicate:

A group address with numbers and letters

AI-generated content may be incorrect.

Fun Fact: In order to vastly increase the performance of my Personal Computer accessing the Virtual/Physical machines over the network, I adjusted my Paging file swap space on my main fast SSD to boost performance for my 16GB PC: A screenshot of a computer

AI-generated content may be incorrect.

This way, more short-term memory can be swapped on the faster 5.15/4.9 read/write GB/s SSD speeds), and can match even beyond the RAM speed.

**A+ Upgrades can play a big role in both networking & cybersecurity!**