1.6 Configure and verify IPv4 addressing and subnetting

To configure and verify IPv4 addressing and subnetting on a Cisco router, you can follow these steps:

1. Connect to your router: Use a console cable to connect to your Cisco router.
2. Enter the user exec mode: Once connected, you can enter the user exec mode by typing `enable` in the command line interface (CLI) and press `Enter`.
3. Enter global configuration mode: To configure the router, you need to enter the global configuration mode. Type `configure terminal` and press `Enter`.
4. Select the interface you want to configure: Type `interface FastEthernet 0/0` (or the appropriate interface you want to use) and press `Enter`.
5. Configure the IP address and subnet mask for the interface: Type `ip address 192.168.1.1 255.255.255.0` (replace 192.168.1.1 and 255.255.255.0 with your desired IP address and subnet mask) and press `Enter`.
6. Enable the interface: Type `no shutdown` and press `Enter`.
7. Exit interface configuration mode: Type `exit` and press `Enter`.
8. Repeat steps 4-7 for each interface you want to configure.
9. Save your changes: To save your changes, type `write memory` or `copy running-config startup-config` and press `Enter`.

To verify your configuration:

1. View the IP configuration: In the privileged exec mode (type `enable` and press `Enter` if you're not already in this mode), type `show ip interface brief`. This command will display a summary of the device's interfaces, including their IP addresses and status.
2. Check the routing table: Type `show ip route` to display the routing table and verify that the network for each interface is listed.

Please replace the IP addresses and subnet mask in the steps with the ones suitable for your network configuration.

1.8 Configure and verify IPv6 addressing and prefix

Configuring and verifying IPv6 addressing and prefix on a Cisco router involves similar steps as with IPv4, but with a few different commands. Here's how to do it:

1. Connect to your router: Use a console cable to connect to your Cisco router.
2. Enter the user exec mode: Once connected, you can enter the user exec mode by typing `enable` in the command line interface (CLI) and press `Enter`.
3. Enter global configuration mode: To configure the router, type `configure terminal` and press `Enter`.
4. Select the interface you want to configure: Type `interface FastEthernet 0/0` (or the appropriate interface you want to use) and press `Enter`.
5. Enable IPv6 processing on the interface: Type `ipv6 enable` and press `Enter`.

6. Configure the IPv6 address and prefix for the interface: Type `ipv6 address 2001:0db8:85a3:0000:0000:8a2e:0370:7334/64` (replace with your desired IPv6 address and prefix length) and press `Enter`.

7. Enable the interface: Type `no shutdown` and press `Enter`.

8. Exit interface configuration mode: Type `exit` and press `Enter`.

9. Repeat steps 4-8 for each interface you want to configure.

10. Save your changes: To save your changes, type `write memory` or `copy running-config startup-config` and press `Enter`.

To verify your configuration:

1. View the IPv6 configuration: In the privileged exec mode (type `enable` and press `Enter` if you're not already in this mode), type `show ipv6 interface brief`. This command will display a summary of the device's interfaces, including their IPv6 addresses and status.

2. Check the IPv6 routing table: Type `show ipv6 route` to display the IPv6 routing table and verify that the network for each interface is listed.

Please replace the IPv6 addresses and prefix in the steps with the ones suitable for your network configuration.

2.1 Configure and verify VLANs (normal range) spanning multiple switches:

2.1.a Access ports (data and voice):

Sure, here are the steps to configure and verify VLANs spanning multiple switches and configuring access ports for data and voice:

1. Connect to your first switch using a console cable and open your terminal program.

2. Once connected, enter the user exec mode by typing `enable` and press `Enter`.

3. Enter the global configuration mode by typing `configure terminal` and press `Enter`.

4. Create a new VLAN by typing `vlan [VLAN_NUMBER]` (replace `[VLAN_NUMBER]` with your desired VLAN number between 1 and 1005). Then press `Enter`.

5. You can optionally give the VLAN a name by typing `name [VLAN_NAME]` (replace `[VLAN_NAME]` with your desired VLAN name) and press `Enter`.

6. Exit back to the global configuration mode by typing `exit` and press `Enter`.

7. Assign a switchport to the VLAN by typing `interface FastEthernet0/1` (replace `FastEthernet0/1` with your desired port), then press `Enter`, followed by `switchport mode access` and `switchport access vlan [VLAN_NUMBER]`.

8. Configure the port for voice VLAN by typing `switchport voice vlan [VOICE_VLAN_NUMBER]`.

9. Repeat these steps for each port you want to add to the VLAN.

10. Save your changes by typing `do write` or `do copy running-config startup-config`.

To verify your configuration:

1. In the privileged exec mode (type `enable` and press `Enter` if you're not already in this mode), type `show vlan brief`. This command will display a summary of the VLANs on the switch.

2. Type `show interfaces status` to see the status and VLAN assignment of each port.

These steps need to be repeated on each switch where you want the VLAN to exist. For the VLAN to span multiple switches, the switches need to be interconnected with trunk ports, which carry traffic from all VLANs by default.

Remember to replace `[VLAN_NUMBER]` with the VLAN number you chose, `[VLAN_NAME]` with the name you want to give the VLAN, `FastEthernet0/1` with the port you want to use, and `[VOICE_VLAN_NUMBER]` with the number of your voice VLAN.

2.1.a.1 Sample config for the access port vlan with Router and Switch:

Below are sample configurations for setting up an access port with VLAN on a Cisco switch and ensuring that the router interface connected to this switch can communicate with this VLAN. This example assumes that the VLAN in question is VLAN 10.

Cisco Switch Configuration

1. Enter global configuration mode:

```
enable
configure terminal
```

2. Create VLAN 10 (if it doesn't already exist):

```
vlan 10
name DataVLAN
exit
```

3. Assign an access port (assuming we're using port FastEthernet0/1) to VLAN 10:

```
interface FastEthernet0/1
switchport mode access
switchport access vlan 10
spanning-tree portfast
exit
```

4. (Optional) If you need to configure an IP address on the switch for VLAN management:

```
interface Vlan10
ip address 192.168.10.2 255.255.255.0
no shutdown
exit
```

5. Save the configuration:

```
write memory
```

- vlan 10 creates VLAN 10 if it doesn't already exist.
- name Data_VLAN assigns a name to the VLAN for easier identification.
- The interface command is used to enter the interface configuration mode for FastEthernet0/1. The actual interface name might vary depending on the model and available interfaces on the switch (e.g., GigabitEthernet1/0/1).

- switchport mode access sets the port to access mode, which means it will carry traffic for only one VLAN.
- switchport access vlan 10 assigns VLAN 10 to the port.
- spanning-tree portfast is typically used on access ports to expedite the transition of the port into the forwarding state, bypassing the usual listening/learning states of Spanning Tree (use with caution and only on edge ports).
- no shutdown enables the interface if it is administratively down.

Cisco Router Configuration

Now, you'll need to configure the router's interface to be able to communicate with VLAN 10. If the router is directly connected to the switch on an access port belonging to VLAN 10, you would configure the router's interface with an IP address in the same subnet as VLAN 10. However, routers typically connect to switches using trunk ports. Here's how you can configure a router with subinterfaces for inter-VLAN routing (Router-on-a-Stick configuration):

1. Enter global configuration mode:

```
enable
configure terminal
```

2. Create a subinterface for VLAN 10 on the router (assuming the router is connected to the switch using GigabitEthernet0/0):

```
interface GigabitEthernet0/0.10
encapsulation dot1Q 10
ip address 192.168.10.1 255.255.255.0
no shutdown
exit
```

3. (Optional) If the router is connected to the switch on an access port and you just need a single VLAN:

```
interface GigabitEthernet0/0
ip address 192.168.10.1 255.255.255.0
no shutdown
exit
```

4. Save the configuration:

```
write memory
```

In the router configuration, `GigabitEthernet0/0.10` is a subinterface for VLAN 10, and `encapsulation dot1Q 10` tells the router to tag the traffic for VLAN 10. The IP address `192.168.10.1` is an example; make sure to use an IP address appropriate for your network. The `no shutdown` command enables the interface.

Please note that these configurations are for Cisco IOS and might be slightly different depending on the switch or router model and the IOS version. Always consult your device documentation for the exact commands.

2.1.a.2 Sample config for the access port voice vlan with Router and Switch:

To configure an access port with a Voice VLAN on a Cisco switch and to ensure that the router interface connected to this switch can communicate with the Voice VLAN, you will need to configure both the switch and router interfaces accordingly.

Cisco Switch Configuration

1. Enter global configuration mode:

```
enable
configure terminal
```

2. Create the Voice VLAN (assuming VLAN 20 is the Voice VLAN):

```
vlan 20
name VoiceVLAN
exit
```

3. Assign an access port (for example, port FastEthernet0/2) to both Data VLAN (VLAN 10) and Voice VLAN (VLAN 20):

```
interface FastEthernet0/2
switchport mode access
switchport access vlan 10
switchport voice vlan 20
spanning-tree portfast
exit
```

```
The `switchport mode access` command sets the port to access mode, the
`switchport access vlan 10` command assigns VLAN 10 as the data VLAN, and the
`switchport voice vlan 20` command specifies VLAN 20 as the Voice VLAN.
```

4. (Optional) Enable Quality of Service (QoS) if it is not already enabled. QoS is often recommended for voice traffic to ensure proper prioritization:

```
mls qos
```

5. Save the configuration:

```
write memory
```

Cisco Router Configuration

For the router, if you are using subinterfaces for inter-VLAN routing (Router-on-a-Stick configuration), you would set up a subinterface for the Voice VLAN as follows:

1. Enter global configuration mode:

```
enable
configure terminal
```

2. Create a subinterface for the Voice VLAN on the router (assuming the router’s interface connected to the switch is GigabitEthernet0/1):

```
interface GigabitEthernet0/1.20
encapsulation dot1Q 20
ip address 192.168.20.1 255.255.255.0
no shutdown
exit
```

```
Here `GigabitEthernet0/1.20` is a subinterface for VLAN 20, and `encapsulation
dot1Q 20` configures the router to recognize traffic tagged with VLAN 20. The IP
address `192.168.20.1` should be in the same subnet as the Voice VLAN and should
be unique within that subnet.
```

3. Save the configuration:

```
write memory
```

This setup will allow the switch port to carry both data and voice traffic, with the voice traffic tagged for VLAN 20. The router will route traffic for VLAN 20 using the subinterface configuration.

Please note that the actual commands and configuration may vary based on your Cisco device model, the IOS version it is running, and the specific requirements of your network. Always refer to the official Cisco documentation or consult with a network professional for configurations specific to your environment.

2.1.b Default VLAN:

Configuring and verifying VLANs spanning multiple switches for the default VLAN involves the following steps:

By default, all switch ports are part of VLAN 1. So if you wish to change the default VLAN, you have to create a new VLAN and assign all ports to the new VLAN.

1. Connect to your switch using a console cable and open your terminal program.

2. Enter the user exec mode by typing `enable` and press `Enter`.

3. Enter the global configuration mode by typing `configure terminal` and press `Enter`.

4. Create a new VLAN by typing `vlan [VLAN_NUMBER]` (replace `[VLAN_NUMBER]` with your desired VLAN number between 1 and 1005). Press `Enter`.

5. Exit back to the global configuration mode by typing `exit` and press `Enter`.

6. Now, assign all ports to the new VLAN by typing `interface range FastEthernet 0/1 - 24` (or the range of your specific model), then `switchport mode access`, then `switchport access vlan [VLAN_NUMBER]` and press `Enter`.

7. Save your changes by typing `exit`, then `copy running-config startup-config`.

To verify your configuration:

1. In the privileged exec mode (type `enable` and press `Enter` if you're not already in this mode), type `show vlan brief`. This command will display a summary of the VLANs on the switch.

2. Type `show interfaces switchport` to see the access mode and VLAN assignment of each port.

Repeat these steps on each switch where you want the VLAN to exist. For the VLAN to span multiple switches, the switches need to be interconnected with trunk ports, which carry traffic from all VLANs by default.

Remember to replace `[VLAN_NUMBER]` with the VLAN number you chose, and adjust the range of interfaces based on your specific switch model.

Sample Config for a default VLAN on Cisco switch and a router:

In a Cisco switch, VLAN 1 is the default VLAN that all switch ports are a part of until they are assigned to a different VLAN. Here's how you can configure the switch and router to communicate using the default VLAN:

Cisco Switch Configuration

1. Enter global configuration mode:

```
enable
configure terminal
```

2. Assign an IP address to the default VLAN (VLAN 1) for management purposes (optional):

```
interface Vlan1
ip address 192.168.1.2 255.255.255.0
no shutdown
exit
```

3. Configure an access port (e.g., FastEthernet0/1) to be part of the default VLAN, which is VLAN 1 by default, so this step may be skipped unless the port has been previously changed:

```
interface FastEthernet0/1
switchport mode access
switchport access vlan 1
spanning-tree portfast
exit
```

4. Save the configuration:

```
write memory
```

Cisco Router Configuration

If you're using a router to communicate with devices in VLAN 1, you can configure the router's interface connected to the switch as follows:

1. Enter global configuration mode:

```
enable
configure terminal
```

2. Configure the router's interface (e.g., GigabitEthernet0/0) with an IP address in the same subnet as the switch's VLAN 1 interface:

```
interface GigabitEthernet0/0
ip address 192.168.1.1 255.255.255.0
no shutdown
exit
```

3. Save the configuration:

```
write memory
```

In this configuration, the switch and router are on the same subnet (192.168.1.0/24), allowing them to communicate. Devices connected to the switch on ports assigned to VLAN 1 will also be able to communicate with the router and any other devices on VLAN 1.

Remember that the default VLAN (VLAN 1) should not be used for carrying sensitive or production traffic because it is often the target of VLAN hopping attacks. It's a best practice to change the management VLAN to a non-default VLAN and ensure that the default VLAN is not used for regular network traffic.

2.1.c InterVLAN connectivity

InterVLAN connectivity allows devices on different VLANs to communicate with each other. This requires a router or a layer 3 switch. Here's how to configure it:

1. Connect to your switch using a console cable and open your terminal program.
2. Enter the user exec mode by typing `enable` and press `Enter`.
3. Enter the global configuration mode by typing `configure terminal` and press `Enter`.
4. Create your VLANs (if not created already) by typing `vlan [VLAN_NUMBER]` (replace `[VLAN_NUMBER]` with your desired VLAN number between 1 and 1005). Press `Enter` and then type `exit`.
5. Assign the VLANs to the ports by typing `interface [INTERFACE_ID]` (replace `[INTERFACE_ID]` with the interface you want to assign), then `switchport mode access` and then `switchport access vlan [VLAN_NUMBER]`.
6. Connect the switches to the router or Layer 3 switch. The interfaces on the router or Layer 3 switch connected to the switches should be configured as sub-interfaces, one for each VLAN.
7. On the router or Layer 3 switch, enter the interface configuration mode by typing `interface [INTERFACE_ID].[VLAN_NUMBER]` (replace `[INTERFACE_ID]` with the interface connected to the switch and `[VLAN_NUMBER]` with the VLAN number).
8. Assign each sub-interface to a VLAN by typing `encapsulation dot1Q [VLAN_NUMBER]` and assign an IP address to each sub-interface by typing `ip address [IP_ADDRESS] [SUBNET_MASK]`.
9. Save your changes by typing `exit`, then `copy running-config startup-config`.

To verify your configuration:

1. On the switch, in the privileged exec mode (type `enable` and press `Enter` if you're not already in this mode), type `show vlan brief`. This command will display a summary of the VLANs on the switch.
2. On the router or Layer 3 switch, type `show ip interface brief` to display the IP addresses of the sub-interfaces.

Remember to replace `[VLAN_NUMBER]`, `[INTERFACE_ID]`, `[IP_ADDRESS]`, and `[SUBNET_MASK]` with your specific VLAN numbers, interfaces, IP addresses, and subnet masks.

Sample config for InterVLAN connectivity on Cisco Switch and Router

To set up InterVLAN connectivity on a Cisco switch and router, you'll typically use a router-on-a-stick configuration. This involves configuring the router with subinterfaces for each VLAN, and the switch with VLANs and trunking to pass traffic for multiple VLANs through a single interface. Below is a sample configuration for a setup with two VLANs (VLAN 10 and VLAN 20) on the switch and a router with two subinterfaces for these VLANs.

Cisco Switch Configuration

1. Enter global configuration mode:

```
enable
configure terminal
```

2. Create the VLANs:

```
vlan 10
name DataVLAN
exit

vlan 20
name VoiceVLAN
exit
```

3. Assign switch ports to VLANs (assuming FastEthernet0/1 for VLAN 10 and FastEthernet0/2 for VLAN 20):

```
interface FastEthernet0/1
switchport mode access
switchport access vlan 10
spanning-tree portfast
exit

interface FastEthernet0/2
switchport mode access
switchport access vlan 20
spanning-tree portfast
exit
```

4. Configure the trunk port on the switch (assuming FastEthernet0/24 connects to the router):

```
interface FastEthernet0/24
switchport mode trunk
switchport trunk allowed vlan 10,20
exit
```

5. Save the configuration:

```
write memory
```

Cisco Router Configuration

1. Enter global configuration mode:

```
enable
configure terminal
```

2. Create subinterfaces for each VLAN on the router's interface connected to the switch trunk port (assuming GigabitEthernet0/0 connects to the switch):

```
interface GigabitEthernet0/0
no shutdown
exit

interface GigabitEthernet0/0.10
encapsulation dot1Q 10
ip address 192.168.10.1 255.255.255.0
no shutdown
exit

interface GigabitEthernet0/0.20
encapsulation dot1Q 20
ip address 192.168.20.1 255.255.255.0
no shutdown
exit
```

3. Save the configuration:

```
write memory
```

In this example, the router's GigabitEthernet0/0 interface has two subinterfaces: GigabitEthernet0/0.10 for VLAN 10 and GigabitEthernet0/0.20 for VLAN 20. The command `encapsulation dot1Q` followed by the VLAN ID tells the router to tag the traffic for that particular VLAN. The IP addresses are the default gateways for the devices in their respective VLANs.

This configuration will allow devices in VLAN 10 to communicate with devices in VLAN 20 and vice versa through the router. Devices will use the router's subinterface IP addresses as their default gateway to reach other networks, including other VLANs and the internet (assuming proper routing is configured on the router).

2.2 Configure and verify interswitch connectivity

2.2.a Trunk ports

Sure, here's how you can configure and verify trunk ports for interswitch connectivity on a Cisco switch:

1. Connect to your switch using a console cable and open your terminal program (like PuTTY).

2. Once connected, enter the user exec mode by typing `enable` and press `Enter`.

3. Enter the global configuration mode by typing `configure terminal` and press `Enter`.

4. Select the interface you want to configure as a trunk port: Type `interface FastEthernet 0/1` (replace `FastEthernet 0/1` with the interface you want to use) and press `Enter`.

5. Set the switchport mode to trunk: Type `switchport mode trunk` and press `Enter`.

6. (Optional) By default, a trunk port will carry all VLANs. If you want to limit which VLANs can be carried over the trunk link, use the `switchport trunk allowed vlan` command followed by the VLAN numbers.

7. Exit the interface configuration mode: Type `exit` and press `Enter`.

8. Save your changes: Type `copy running-config startup-config` and press `Enter`.

To verify your trunk configuration:

1. In the privileged exec mode (type `enable` and press `Enter` if you're not already in this mode), type `show interfaces trunk`. This command will display information about all the current trunk ports including the native VLAN and the allowed VLANs on each trunk.

Please replace `FastEthernet 0/1` with the interface you want to configure as a trunk port.

Sample config for Trunk ports on Cisco devices

Trunk ports on Cisco switches are used to carry VLAN traffic between switches and other network devices like routers or other switches. Here's an example of how to configure a trunk port on a Cisco switch.

Cisco Switch Trunk Port Configuration

1. Access the switch via the console or SSH and enter global configuration mode:

```
enable
configure terminal
```

2. Define the VLANs that will be allowed across the trunk (unless they are already defined):

```
vlan 10
name Marketing
exit

vlan 20
name Engineering
exit

vlan 30
name HR
exit
```

3. Now, configure the trunk port. Assume we are using interface GigabitEthernet0/1 as the trunk port:

```
interface GigabitEthernet0/1
switchport mode trunk
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 10,20,30
```

  - `switchport mode trunk` sets the port to trunk mode.
  - `switchport trunk encapsulation dot1q` sets the trunk to use IEEE 802.1Q frame tagging. Note that this command is only necessary on switches that support both ISL and dot1Q and is not present on newer switches that only support dot1Q.
  - `switchport trunk allowed vlan 10,20,30` specifies which VLANs are allowed to pass through the trunk port. If you want all VLANs to pass through the trunk, you can use `switchport trunk allowed vlan all` or simply omit this command since all VLANs are allowed on the trunk by default.

4. (Optional) It's a good practice to set the native VLAN to a VLAN that is not used for normal network traffic to mitigate VLAN hopping attacks. The native VLAN is the one that carries untagged traffic:

   ```
   switchport trunk native vlan 999
   ```

5. Save the configuration:

   ```
   write memory
   ```

Cisco Router Trunk Port Configuration

Typically, routers do not have 'trunk' ports in the same way switches do. However, if you're using a router with switch modules or want to configure a router's interface to route multiple VLANs using subinterfaces, you might do something like this:

1. Enter global configuration mode:

   ```
   enable
   configure terminal
   ```

2. Configure subinterfaces for each VLAN (assuming GigabitEthernet0/0 connects to the switch's trunk port):

   ```
   interface GigabitEthernet0/0
   no shutdown
   ```

   For VLAN 10:

   ```
   interface GigabitEthernet0/0.10
   encapsulation dot1Q 10
   ip address 192.168.10.1 255.255.255.0
   no shutdown
   ```

   For VLAN 20:

   ```
   interface GigabitEthernet0/0.20
   encapsulation dot1Q 20
   ip address 192.168.20.1 255.255.255.0
   no shutdown
   ```

   And so on for additional VLANs. Each subinterface's IP address should be in the corresponding VLAN's subnet and will act as the default gateway for devices in that VLAN.

3. Save the configuration:

   ```
   write memory
   ```

Please ensure you're configuring the correct interfaces as per your network topology and replace IP addresses and VLAN IDs with those relevant to your network. Also, newer switches might not require the encapsulation command as they might only support 802.1Q and not ISL. Always check the documentation for your specific Cisco device model and software version.

2.2.b 802.1Q

The 802.1Q protocol is used for VLAN tagging in Ethernet networks. This protocol allows multiple VLANs to be used over a single link (trunk link) between switches. Here's how to configure and verify it on a Cisco switch:

1. Connect to your switch using a console cable and open your terminal program (like PuTTY).
2. Once connected, enter the user exec mode by typing `enable` and press `Enter`.
3. Enter the global configuration mode by typing `configure terminal` and press `Enter`.
4. Select the interface you want to configure as a trunk port: Type `interface FastEthernet 0/1` (replace `FastEthernet 0/1` with the interface you want to use) and press `Enter`.
5. Set the switchport mode to trunk: Type `switchport mode trunk` and press `Enter`.
6. Set the encapsulation mode to 802.1Q: Type `switchport trunk encapsulation dot1q` and press `Enter`.
7. (Optional) By default, a trunk port will carry all VLANs. If you want to limit which VLANs can be carried over the trunk link, use the `switchport trunk allowed vlan` command followed by the VLAN numbers.
8. Exit the interface configuration mode: Type `exit` and press `Enter`.
9. Save your changes: Type `copy running-config startup-config` and press `Enter`.

To verify your trunk configuration:

1. In the privileged exec mode (type `enable` and press `Enter` if you're not already in this mode), type `show interfaces trunk`. This command will display information about all the current trunk ports including the encapsulation type (should be 802.1Q), native VLAN and the allowed VLANs on each trunk.

Please replace `FastEthernet 0/1` with the interface you want to configure as a trunk port.

Sample config for 802.1Q ports on Cisco devices

Configuring 802.1Q on Cisco devices involves setting up VLANs and configuring trunk ports to use 802.1Q encapsulation to carry traffic for multiple VLANs across a single physical link.

Here's a sample configuration for setting up 802.1Q trunking on Cisco switches and routers:

Cisco Switch Configuration for 802.1Q Trunking

1. Enter global configuration mode:

   ```
   enable
   configure terminal
   ```

2. (Optional) Define VLANs that will be carried by the trunk port:

```
vlan 10
name DataVLAN
exit

vlan 20
name VoiceVLAN
exit
```

3. Configure the trunk port (assuming we're using interface GigabitEthernet1/0/1):

```
interface GigabitEthernet1/0/1
switchport mode trunk
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 10,20
```

   - `switchport mode trunk` sets the interface to trunk mode.
   - `switchport trunk encapsulation dot1q` specifies that the trunk will use 802.1Q VLAN tagging. This command is necessary on some older switches but might not be available or required on newer models that only support dot1Q.
   - `switchport trunk allowed vlan 10,20` defines which VLANs are allowed on the trunk. By default, all VLANs are allowed.

4. Save the configuration:

```
write memory
```

Cisco Router Configuration for 802.1Q Trunking

Configuring 802.1Q trunking on a router typically means setting up subinterfaces for each VLAN that will be routed by the router. This is often used in a router-on-a-stick configuration:

1. Enter global configuration mode:

```
enable
configure terminal
```

2. Create a subinterface for each VLAN on the router's interface connected to the switch trunk port. In this example, let's assume the router's interface is GigabitEthernet0/0:

```
interface GigabitEthernet0/0
no shutdown
```

For VLAN 10:

```
interface GigabitEthernet0/0.10
description DataVLAN
encapsulation dot1Q 10
ip address 192.168.10.1 255.255.255.0
no shutdown
```

For VLAN 20:

```
interface GigabitEthernet0/0.20
description VoiceVLAN
encapsulation dot1Q 20
ip address 192.168.20.1 255.255.255.0
no shutdown
```

The `encapsulation dot1Q VLAN_ID` command is essential as it tells the router which VLAN each subinterface belongs to.

3. Save the configuration:

```
write memory
```

These configurations enable devices on different VLANs to communicate with each other through the router, which will route traffic between the VLANs.

Remember to adjust the interface names, IP addresses, and VLAN numbers to match your network environment. Ensure that the VLANs you are configuring on the router's subinterfaces are allowed on the switch's trunk port.

If the `switchport trunk encapsulation dot1q` command is not accepted, it may mean that the switch only supports 802.1Q and does not need the command, or that you are on an interface that does not support trunking. Always refer to the documentation specific to your Cisco device model and IOS version for detailed configuration guidelines.

2.2.c Native VLAN

The Native VLAN is a specified VLAN that a trunk port sends untagged frames to when it receives untagged frames. Here's how to configure and verify it:

1. Connect to your switch using a console cable and open your terminal program.
2. Once connected, enter the user exec mode by typing `enable` and press `Enter`.
3. Enter the global configuration mode by typing `configure terminal` and press `Enter`.
4. Choose the interface you wish to configure as a trunk port: Type `interface FastEthernet 0/1` (replace `FastEthernet 0/1` with the interface you want to use), and press `Enter`.
5. Set the switchport mode to trunk: Type `switchport mode trunk` and press `Enter`.
6. Set the Native VLAN ID: Type `switchport trunk native vlan [VLAN_ID]` (replace `[VLAN_ID]` with the VLAN ID you want to use as the Native VLAN), and press `Enter`.
7. Exit the interface configuration mode: Type `exit` and press `Enter`.

8. Save your changes: Type `copy running-config startup-config` and press `Enter`.

To verify your Native VLAN configuration:

1. In the privileged exec mode (type `enable` and press `Enter` if you're not already in this mode), type `show interfaces trunk`. This command will display information about all the current trunk ports, including the Native VLAN for each trunk.

Note: For proper network operation, the Native VLAN should be the same on both ends of the trunk link.

Please replace `FastEthernet 0/1` and `[VLAN_ID]` with the interface and VLAN ID you want to configure as the Native VLAN, respectively.

Sample config for native vlan ports on Cisco devices

In an 802.1Q trunk, the native VLAN is used for untagged traffic. By default, the native VLAN on a Cisco switch is VLAN 1, but it's a best practice to change this to a different VLAN for security purposes. Here's how you can configure the native VLAN on a trunk port of a Cisco switch:

Cisco Switch Configuration for Native VLAN

1. Access the switch via the console or SSH and enter global configuration mode:

```
enable
configure terminal
```

2. (Optional) Create a new VLAN to be used as the native VLAN if it doesn't already exist (assuming we are using VLAN 99):

```
vlan 99
name NativeVLAN
exit
```

3. Configure the trunk port and set the native VLAN (assuming the trunk port is GigabitEthernet1/0/1):

```
interface GigabitEthernet1/0/1
switchport mode trunk
switchport trunk encapsulation dot1q  # This command may be omitted on newer switches
switchport trunk native vlan 99
switchport trunk allowed vlan 10,20,99  # Assuming VLANs 10 and 20 are also allowed on this trunk
```

4. Ensure that the native VLAN is the same on both ends of the trunk link.

5. (Optional) It's a good security practice to ensure the native VLAN is not assigned to any switch ports:

```
interface range GigabitEthernet1/0/2 - 48
switchport mode access
switchport access vlan 10  # Assign an access VLAN other than the native VLAN
```

6. Save the configuration:

```
write memory
```

Explanation:

- `switchport mode trunk` sets the interface to trunk mode.
- `switchport trunk encapsulation dot1q` specifies that the trunk should use IEEE 802.1Q frame tagging. Some newer models only support 802.1Q and do not require this command.
- `switchport trunk native vlan 99` sets VLAN 99 as the native VLAN for the trunk port.
- `switchport trunk allowed vlan 10,20,99` specifies which VLANs are allowed on the trunk. Including the native VLAN in the allowed list is optional since the native VLAN is allowed on the trunk by default.
- The `interface range` command is used to configure multiple interfaces at once. Replace `GigabitEthernet1/0/2 – 48` with the actual range of your switch ports.

Cisco Router Configuration

Routers do not have a concept of a native VLAN on their physical interfaces, but if you are using a router-on-a-stick configuration with subinterfaces, you can configure one of the subinterfaces to handle untagged traffic (effectively acting as the native VLAN):

1. Enter global configuration mode:

```
enable
configure terminal
```

2. Configure the physical interface connected to the switch trunk port to be up:

```
interface GigabitEthernet0/0
no shutdown
```

3. Create a subinterface for the native VLAN (assuming the native VLAN is 99):

```
interface GigabitEthernet0/0.99
encapsulation dot1Q 99 native
ip address 192.168.99.1 255.255.255.0
no shutdown
```

Here, `encapsulation dot1Q 99 native` tells the router that this subinterface is for the native VLAN (VLAN 99), and traffic for this VLAN will not be tagged.

4. Save the configuration:

```
write memory
```

Remember to replace the interface names, VLAN IDs, and IP addresses with those relevant to your network setup. Ensure that you have matching native VLAN configurations on both the switch and the router (or another switch) to avoid VLAN mismatch issues.

2.3 Configure and verify Layer 2 discovery protocols (Cisco Discovery Protocol and LLDP)

Sure, here's how you can configure and verify Layer 2 discovery protocols like Cisco Discovery Protocol (CDP) and Link Layer Discovery Protocol (LLDP) on a Cisco switch:

1. Connect to your switch using a console cable and open your terminal program.

2. Once connected, enter the user exec mode by typing `enable` and press `Enter`.

3. Enter the global configuration mode by typing `configure terminal` and press `Enter`.

For CDP:

1. To enable CDP globally, type `cdp run` and press `Enter`.

2. To disable CDP on a specific interface, first select the interface by typing `interface FastEthernet 0/1` (replace `FastEthernet 0/1` with the interface you want), press `Enter`, then type `no cdp enable`.

For LLDP:

1. To enable LLDP globally, type `lldp run` and press `Enter`.

2. To disable LLDP on a specific interface, select the interface by typing `interface FastEthernet 0/1` (replace `FastEthernet 0/1` with the interface you want), press `Enter`, then type `no lldp transmit` and `no lldp receive`.

To verify your configuration:

1. For CDP, in the privileged exec mode (type `enable` and press `Enter` if you're not already in this mode), type `show cdp neighbors` to display a summary of neighboring devices discovered by CDP.

2. For LLDP, in the privileged exec mode, type `show lldp neighbors` to display a summary of neighboring devices discovered by LLDP.

Please replace `FastEthernet 0/1` with the interface you want to configure.

Sample configure and verify Layer 2 discovery protocols CDP:

Cisco Discovery Protocol (CDP) is a Cisco proprietary Layer 2 network discovery protocol that is enabled by default on most Cisco devices. CDP allows network devices to share information about themselves with directly connected devices. Here's how you can configure and verify CDP on a Cisco device:

Configuring CDP on a Cisco Device

1. Connect to your Cisco device and enter global configuration mode:

   ```
   enable
   configure terminal
   ```

2. To enable CDP globally (if it's not already enabled), use:

   ```
   cdp run
   ```

3. To disable CDP globally, use:

   ```
   no cdp run
   ```

4. To enable CDP on a specific interface (if it has been disabled), go into the interface configuration mode and use:

   ```
   interface [interface_type/number]
   cdp enable
   ```

   Replace `[interface_type/number]` with the actual interface identifier, such as `GigabitEthernet0/1`.

5. To disable CDP on a specific interface, use:

   ```
   interface [interface_type/number]
   no cdp enable
   ```

6. Save the configuration:

   ```
   write memory
   ```

Verifying CDP Configuration

1. To check the CDP status globally, use the following command:

   ```
   show cdp
   ```

   This command displays the CDP timer and hold-time information, as well as whether CDP is enabled globally.

2. To see information about neighbors discovered by CDP, use:

   ```
   show cdp neighbors
   ```

   This command provides details about directly connected Cisco devices, including device identifiers, local and remote interfaces, capabilities, platform, and the hold time value.

3. For detailed information about a specific neighbor, including network addresses and more, use:

   ```
   show cdp neighbors [interface_type/number] detail
   ```

   Replace `[interface_type/number]` with the specific interface you want to check.

4. To view the CDP configuration for a specific interface, use:

   ```
   show cdp interface [interface_type/number]
   ```

   This command shows whether CDP is enabled on the interface and the specific CDP settings for that interface.

Remember that CDP is a Cisco proprietary protocol and only works with Cisco devices or devices from vendors that have implemented CDP. It is not encrypted and can pose a security risk if used on interfaces facing untrusted networks. Therefore, it’s generally recommended to disable CDP on such interfaces.

Sample configure and verify Layer 2 discovery protocols LLDP:

Link Layer Discovery Protocol (LLDP) is an industry-standard Layer 2 protocol that allows devices to advertise and discover connected devices and their capabilities. Unlike CDP, which is Cisco proprietary, LLDP works across different vendors' equipment. Here's how to configure and verify LLDP on a Cisco device:

Configuring LLDP on a Cisco Device

1. Connect to your Cisco device and enter global configuration mode:

   ```
   enable
   configure terminal
   ```

2. To enable LLDP globally (if it's not already enabled), use:

   ```
   lldp run
   ```

3. To disable LLDP globally, use:

   ```
   no lldp run
   ```

4. To enable LLDP on a specific interface (if it has been disabled), go into the interface configuration mode and use:

   ```
   interface [interface_type/number]
   lldp transmit
   lldp receive
   ```

   Replace `[interface_type/number]` with the actual interface identifier, such as `GigabitEthernet0/1`.

5. To disable LLDP on a specific interface, use:

   ```
   interface [interface_type/number]
   no lldp transmit
   no lldp receive
   ```

6. Save the configuration:

   ```
   write memory
   ```

Verifying LLDP Configuration

1. To check the LLDP status globally, use the following command:

   ```
   show lldp
   ```

   This command displays whether LLDP is enabled globally and the timers for LLDP advertisements.

2. To see information about neighbors discovered by LLDP, use:

   ```
   show lldp neighbors
   ```

   This command provides details about directly connected devices that are LLDP-capable, including the local and remote interfaces, port IDs, and system names.

3. For detailed information about a specific neighbor, including capabilities and network addresses, use:

   ```
   show lldp neighbors [interface_type/number] detail
   ```

   Replace `[interface_type/number]` with the specific interface you want to check.

4. To view the LLDP configuration for a specific interface, use:

   ```
   show lldp interface [interface_type/number]
   ```

   This command shows whether LLDP is enabled for transmission and reception on the interface and the specific LLDP settings for that interface.

Remember to replace the interface names with those relevant to your network setup. LLDP can be useful for network discovery and troubleshooting in multi-vendor environments. It's generally safe to enable LLDP across the network because it facilitates better visibility of the network topology and connected devices. However, like any discovery protocol, consider the security implications of advertising device information and use it judiciously on interfaces that face untrusted networks.

2.4 Configure and verify (Layer 2/Layer 3) EtherChannel (LACP):

EtherChannel allows you to combine several physical Ethernet links to create one logical Ethernet link for the purpose of providing fault-tolerance and high-speed links between switches, routers, and servers. Here's how to configure and verify it:

1. Connect to your switch using a console cable and open your terminal program.

2. Once connected, enter the user exec mode by typing `enable` and press `Enter`.

3. Enter the global configuration mode by typing `configure terminal` and press `Enter`.

4. Choose the interfaces you want to configure as part of the EtherChannel. Type `interface range FastEthernet 0/1 - 2` (replace `FastEthernet 0/1 - 2` with the interfaces you want to use), and press `Enter`.

5. Set the interfaces to use LACP (Link Aggregation Control Protocol) by typing `channel-group 1 mode active` and press `Enter`. This will create a new port-channel interface (Port-channel 1) and enable LACP.

6. Exit the interface configuration mode: Type `exit` and press `Enter`.

7. Now, configure the Port-channel interface. Type `interface Port-channel 1` and press `Enter`.

8. (Optional) If you are configuring Layer 3 EtherChannel, assign an IP address to the Port-channel interface by typing `ip address [IP_ADDRESS] [SUBNET_MASK]` and press `Enter`.

9. Save your changes: Type `exit`, then `copy running-config startup-config` and press `Enter`.

To verify your EtherChannel configuration:

1. In the privileged exec mode (type `enable` and press `Enter` if you're not already in this mode), type `show etherchannel summary`. This command will display information about the EtherChannel, including the interfaces in the channel and the protocol used.

Please replace `FastEthernet 0/1 - 2`, `[IP_ADDRESS]`, and `[SUBNET_MASK]` with the interfaces, IP address, and subnet mask you want to use, respectively.

Sample configuration for the Layer 2/Layer 3 EtherChannel LACP:

Link Aggregation Control Protocol (LACP) is part of an IEEE specification (802.3ad) that allows several physical ports to be bundled to form a single logical channel. EtherChannel is the name Cisco uses for the method that allows multiple physical Ethernet links to combine into one logical channel. This can be done in both Layer 2 (switching) and Layer 3 (routing) contexts. Here's how to configure LACP for EtherChannel on Cisco devices:

Sample Configuration for Layer 2 LACP EtherChannel on a Cisco Switch

1. Enter global configuration mode:

```
enable
configure terminal
```

2. Create the Port-Channel interface (logical interface representing the EtherChannel). Let's use Port-Channel 1 for this example:

```
interface Port-channel1
switchport
switchport mode trunk  # or "switchport access vlan X" for an access port-channel
switchport trunk allowed vlan 10,20,30  # if in trunk mode, specify allowed VLANs
```

3. Add physical interfaces to the EtherChannel. For example, let's bundle GigabitEthernet0/1 and GigabitEthernet0/2:

```
interface range GigabitEthernet0/1 - 2
switchport mode trunk  # Make sure the physical interfaces have the same settings as the Port-Channel
channel-group 1 mode active  # This command enables LACP on the interfaces and adds them to Port-Channel 1
exit
```

The `channel-group 1 mode active` command tells the interfaces to actively negotiate forming an LACP EtherChannel.

4. Save the configuration:

```
write memory
```

Sample Configuration for Layer 3 LACP EtherChannel on a Cisco Router

1. Enter global configuration mode:

```
enable
configure terminal
```

2. Create the Port-Channel interface:

   ```
   interface Port-channel1
   no switchport
   ip address 192.168.1.1 255.255.255.0  # Assign IP address to the logical interface
   ```

3. Add physical interfaces to the EtherChannel. For example, let’s bundle GigabitEthernet0/0/0 and GigabitEthernet0/0/1:

   ```
   interface range GigabitEthernet0/0/0 - 0/0/1
   no switchport  # Make sure the physical interfaces are in Layer 3 mode
   channel-group 1 mode active  # Enable LACP and add them to Port-Channel 1
   exit
   ```

4. Save the configuration:

   ```
   write memory
   ```

Verifying LACP EtherChannel Configuration

After configuring LACP EtherChannel, you can verify that it’s set up correctly:

1. To verify the EtherChannel summary, use:

   ```
   show etherchannel summary
   ```

   This command displays the status of the Port-Channel, including the number of ports in the channel and their statuses.

2. To check detailed information about the EtherChannel, use:

   ```
   show etherchannel 1 port-channel  # Replace '1' with your Port-Channel number
   ```

3. For detailed protocol information, use:

   ```
   show etherchannel 1 detail  # Replace '1' with your Port-Channel number
   ```

Remember to replace the interface names and numbers with the appropriate ones for your environment. Also, ensure that the configurations (like trunk settings or IP addresses) on both sides of the EtherChannel match. It’s worth noting that all interfaces in an EtherChannel must have the same speed and duplex settings.

2.9 Configure the components of a wireless LAN access for client connectivity using GUI only such as WLAN creation, security settings, QoS profiles, and advanced WLAN settings:

To configure the components of a wireless LAN access for client connectivity using GUI only, you’d typically use a web interface provided by the wireless router or access point. The exact steps can vary depending on the specific model of your wireless device, but here’s a general guideline:

1. **Accessing the web interface**:
   Connect to your wireless router or access point either wirelessly or using an Ethernet cable. Open a web browser and enter the IP address of the device in the address bar (often something like 192.168.0.1 or 192.168.1.1). You should be prompted for a username and password.

2. **WLAN creation**:
   Once logged in, look for a section or tab labeled "Wireless," "Wireless Settings," or something similar. Here, you can set up your wireless network(s). You might have the option to create multiple wireless networks (SSIDs).

3. **Security settings**:
   Still under the "Wireless" section, look for "Security" settings. You can select the type of security (WPA2 is generally recommended), and set the password (also known as a pre-shared key) for the network.

4. **QoS profiles**:
   Quality of Service (QoS) settings can usually be found under their own section or tab, labeled "QoS" or something similar. Here, you can prioritize different types of traffic to ensure, for example, that video streaming gets priority over file downloads.

5. **Advanced WLAN settings**:
   Under "Wireless" or a separate "Advanced" section, you can adjust various other settings like the wireless mode (b/g/n/ac), channel width, and more. Unless you know what you're doing, it's often best to leave these at their default settings.

6. **Save your settings**:
   Make sure to save your settings before exiting. There is usually a "Save" or "Apply" button at the bottom of each page.

Remember, these are general steps. The exact labels and locations of these settings can vary widely between different devices, so refer to your device's manual or online documentation for specific instructions.

3.3 Configure and verify IPv4 and IPv6 static routing
3.3.a Default route

Configuring static routing, including a default route, on a Cisco router involves the following steps.

For IPv4:

1. Connect to your router using console cable and launch your terminal program.

2. Enter the user exec mode by typing `enable` in the command line interface (CLI).

3. Enter the global configuration mode by typing `configure terminal`.

4. To set a default route, type `ip route 0.0.0.0 0.0.0.0 [next_hop_IP]`, where `[next_hop_IP]` is the IP address of the next hop router or exit interface.

5. Press `Enter`.

6. Save your changes by typing `exit`, then `copy running-config startup-config`.

For IPv6:

1. Connect to your router using console cable and launch your terminal program.

2. Enter the user exec mode by typing `enable` in the command line interface (CLI).

3. Enter the global configuration mode by typing `configure terminal`.

4. To set a default route, type `ipv6 route ::/0 [next_hop_IPv6]`, where `[next_hop_IPv6]` is the IPv6 address of the next hop router or exit interface.

5. Press `Enter`.

6. Save your changes by typing `exit`, then `copy running-config startup-config`.

To verify your configuration:

1. In the privileged exec mode, type `show ip route` for IPv4 or `show ipv6 route` for IPv6 to display the routing table and verify that the default route is listed.

Please replace `[next_hop_IP]` and `[next_hop_IPv6]` with the IP address of your next hop router or exit interface.

3.3.b Network route

To configure IPv4 and IPv6 static routes on a Cisco router, follow these steps:

For IPv4:

1. Connect to your router using console cable and launch your terminal program.

2. Enter the user exec mode by typing `enable` in the command line interface (CLI).

3. Enter the global configuration mode by typing `configure terminal`.

4. To set a static route, type `ip route [destination_network] [subnet_mask] [next_hop_IP]` or `ip route [destination_network] [subnet_mask] [exit_interface]`, where `[destination_network]` is the network you want to reach, `[subnet_mask]` is the subnet mask of the destination network, `[next_hop_IP]` is the IP address of the next-hop router, and `[exit_interface]` is the local interface to use to forward packets.

5. Press `Enter`.

6. Save your changes by typing `exit`, then `copy running-config startup-config`.

For IPv6:

1. Connect to your router using console cable and launch your terminal program.

2. Enter the user exec mode by typing `enable` in the command line interface (CLI).

3. Enter the global configuration mode by typing `configure terminal`.

4. To set a static route, type `ipv6 route [destination_network] [next_hop_IPv6]` or `ipv6 route [destination_network] [exit_interface]`, where `[destination_network]` is the network you want to reach, `[next_hop_IPv6]` is the IPv6 address of the next-hop router, and `[exit_interface]` is the local interface to use to forward packets.

5. Press `Enter`.

6. Save your changes by typing `exit`, then `copy running-config startup-config`.

To verify your configuration:

1. In the privileged exec mode, type `show ip route` for IPv4 or `show ipv6 route` for IPv6 to display the routing table and verify that the static route is listed.

Please replace `[destination_network]`, `[subnet_mask]`, `[next_hop_IP]`, `[next_hop_IPv6]`, and `[exit_interface]` with your specific network details.

3.3.c Host route

To configure IPv4 and IPv6 static routes for a specific host on a Cisco router, follow these steps:

For IPv4:

1. Connect to your router using console cable and launch your terminal program.
2. Enter the user exec mode by typing `enable` in the command line interface (CLI).
3. Enter the global configuration mode by typing `configure terminal`.
4. To set a static route to a specific host, type `ip route [host_IP] 255.255.255.255 [next_hop_IP]` or `ip route [host_IP] 255.255.255.255 [exit_interface]`, where `[host_IP]` is the IP address of the host you want to reach, `[next_hop_IP]` is the IP address of the next-hop router, and `[exit_interface]` is the local interface to use to forward packets.
5. Press `Enter`.
6. Save your changes by typing `exit`, then `copy running-config startup-config`.

For IPv6:

1. Connect to your router using console cable and launch your terminal program.
2. Enter the user exec mode by typing `enable` in the command line interface (CLI).
3. Enter the global configuration mode by typing `configure terminal`.
4. To set a static route to a specific host, type `ipv6 route [host_IPv6]/128 [next_hop_IPv6]` or `ipv6 route [host_IPv6]/128 [exit_interface]`, where `[host_IPv6]` is the IPv6 address of the host you want to reach, `[next_hop_IPv6]` is the IPv6 address of the next-hop router, and `[exit_interface]` is the local interface to use to forward packets.
5. Press `Enter`.
6. Save your changes by typing `exit`, then `copy running-config startup-config`.

To verify your configuration:

1. In the privileged exec mode, type `show ip route` for IPv4 or `show ipv6 route` for IPv6 to display the routing table and verify that the static route to the host is listed.

Please replace `[host_IP]`, `[next_hop_IP]`, `[host_IPv6]`, `[next_hop_IPv6]`, and `[exit_interface]` with your specific host and network details.

3.3.d Floating static

A floating static route is a static route that has an administrative distance greater than the administrative distance (AD) of another static route or dynamic routes. Floating static routes provide a backup path in case of a failure of the primary path.

Here’s how to configure a floating static route for IPv4 and IPv6:

For IPv4:

1. Connect to your router using a console cable and launch your terminal program.

2. Enter the user exec mode by typing `enable` in the command line interface (CLI).

3. Enter the global configuration mode by typing `configure terminal`.

4. To set a floating static route, type `ip route [destination_network] [subnet_mask] [next_hop_IP] [administrative_distance]`, where `[destination_network]` is the network you want to reach, `[subnet_mask]` is the subnet mask of the destination network, `[next_hop_IP]` is the IP address of the next-hop router, and `[administrative_distance]` is the AD you want to set (it should be higher than the AD of the primary route).

5. Press `Enter`.

6. Save your changes by typing `exit`, then `copy running-config startup-config`.

For IPv6:

1. Connect to your router using a console cable and launch your terminal program.

2. Enter the user exec mode by typing `enable` in the command line interface (CLI).

3. Enter the global configuration mode by typing `configure terminal`.

4. To set a floating static route, type `ipv6 route [destination_network] [next_hop_IPv6] [administrative_distance]`, where `[destination_network]` is the network you want to reach, `[next_hop_IPv6]` is the IPv6 address of the next-hop router, and `[administrative_distance]` is the AD you want to set (it should be higher than the AD of the primary route).

5. Press `Enter`.

6. Save your changes by typing `exit`, then `copy running-config startup-config`.

To verify your configuration:

1. In the privileged exec mode, type `show ip route` for IPv4 or `show ipv6 route` for IPv6 to display the routing table and verify that the floating static route is listed.

Please replace `[destination_network]`, `[subnet_mask]`, `[next_hop_IP]`, `[next_hop_IPv6]`, and `[administrative_distance]` with your specific network details.

Sample config for the IPv4 and IPv6 static routing on Cisco Devices:

Static routing is a simple form of routing, recommended only for small networks or when you need to create a single route or a stub network. Below are sample configurations for both IPv4 and IPv6 static routing on Cisco devices.

IPv4 Static Routing

To configure IPv4 static routing on a Cisco device, you need to define a route for a specific network or default route for all traffic.

1. Connect to the Cisco device and enter global configuration mode:

   ```
   enable
   configure terminal
   ```

2. Add a static route to a specific network:

```
ip route [destination_network] [subnet_mask] [next_hop_address or
exit_interface]
```

Example for a specific network:

```
ip route 192.168.2.0 255.255.255.0 192.168.1.2
```

This tells the router that to reach the network 192.168.2.0/24, it should forward packets to the next hop IP address 192.168.1.2.

3. Add a default static route (gateway of last resort):

```
ip route 0.0.0.0 0.0.0.0 [next_hop_address or exit_interface]
```

Example for a default route:

```
ip route 0.0.0.0 0.0.0.0 GigabitEthernet0/1
```

This configures the router to send all traffic for which it does not have a specific route to the interface GigabitEthernet0/1.

4. Save the configuration:

```
write memory
```

IPv6 Static Routing

IPv6 static routing is quite similar to IPv4. Here's how to configure it:

1. Connect to the Cisco device and enter global configuration mode:

```
enable
configure terminal
```

2. Add a static route to a specific network:

```
ipv6 route [destination_network] [prefix_length] [next_hop_address or
exit_interface]
```

Example for a specific network:

```
ipv6 route 2001:db8:acad:2::/64 2001:db8:acad:1::1
```

This tells the router that to reach the network 2001:db8:acad:2::/64, it should forward packets to the next hop IPv6 address 2001:db8:acad:1::1.

3. Add a default static route (gateway of last resort):

```
ipv6 route ::/0 [next_hop_address or exit_interface]
```

Example for a default route:

```
ipv6 route ::/0 GigabitEthernet0/1
```

This configures the router to send all traffic for which it does not have a specific route to the interface GigabitEthernet0/1.

4. Save the configuration:

```
write memory
```

Verification

After configuring static routing, you can verify the routes with the following commands:

- For IPv4 routes:

```
show ip route
```

- For IPv6 routes:

```
show ipv6 route
```

These commands display the routing table entries, including directly connected networks and static routes that you've configured.

Remember that static routes require manual updates if network changes occur. For larger, more dynamic networks, a dynamic routing protocol like OSPF, EIGRP, or BGP is often more appropriate.

3.4 Configure and verify single area OSPFv2:

3.4.a Neighbor adjacencies

Open Shortest Path First version 2 (OSPFv2) is a routing protocol for IPv4. Here's how to configure OSPFv2 and verify neighbor adjacencies:

1. Connect to your router using a console cable and open your terminal program.

2. Enter the user exec mode by typing `enable` and press `Enter`.

3. Enter the global configuration mode by typing `configure terminal`.

4. Enable the OSPF process and assign a process ID by typing `router ospf [process_id]`, where `[process_id]` is a number you assign to uniquely identify the OSPF process on the router. The range is from 1 to 65535.

5. Assign the router to an OSPF area by typing `network [network_address] [wildcard_mask] area [area_id]`. Here, `[network_address]` is the IP address of the network, `[wildcard_mask]` is the inverse of the subnet mask, and `[area_id]` is the ID of the area to which you want to assign the network.

6. (Optional) Configure the router ID, which helps identify the router in the OSPF network, by typing `router-id [router_id]`, where `[router_id]` is the ID you want to assign to the router. If not configured manually, the router will use the highest IP address of its active interfaces.

7. Save your changes by typing `exit`, then `copy running-config startup-config`.

To verify your configuration:

1. In the privileged exec mode (type `enable` and press `Enter` if you're not already in this mode), type `show ip ospf neighbor`. This command will display information about OSPF neighbor adjacencies, including the neighbor's router ID, priority, state, and IP address.

Please replace `[process_id]`, `[network_address]`, `[wildcard_mask]`, `[area_id]`, and `[router_id]` with your specific OSPF parameters.

Sample Configuration for single area OSPFv2 with Neighbor adjacencies

Open Shortest Path First version 2 (OSPFv2) is used for routing IP traffic in an IPv4 network using link-state information. Below is a sample configuration for a single-area OSPF setup on Cisco devices. For this example, we will configure OSPF in area 0, which is the backbone area.

Sample Configuration for Single-Area OSPFv2

1. Access your Cisco router and enter global configuration mode:

   ```
   enable
   configure terminal
   ```

2. Enable OSPF on the router and assign a process ID (the process ID is locally significant to the router):

   ```
   router ospf 1
   ```

3. Assign the router ID (optional, but recommended for stability):

   ```
   router-id 1.1.1.1
   ```

4. Configure OSPF on the interfaces that will participate in OSPF. For instance, if you want to include interfaces with IP addresses in the 192.168.1.0/24 and 10.0.0.0/8 networks in OSPF, you can do so with the `network` command:

   ```
   network 192.168.1.0 0.0.0.255 area 0
   network 10.0.0.0 0.255.255.255 area 0
   ```

   The wildcard mask (`0.0.0.255` and `0.255.255.255`) is the inverse of the subnet mask, specifying which bits of the IP address must match.

5. (Optional) If you want to explicitly define OSPF settings on an interface level:

   ```
   interface GigabitEthernet0/0
   ip ospf 1 area 0
   ```

6. Ensure that OSPF is configured with the same area ID on adjacent routers for them to form neighbor adjacencies.

7. Save the configuration:

   ```
   write memory
   ```

Verification of OSPF Neighbor Adjacencies

After configuring OSPF, you can verify that neighbor adjacencies have formed with the following commands:

1. To verify OSPF neighbor relationships:

   ```
   show ip ospf neighbor
   ```

   This command displays information about OSPF neighbors, including their state (e.g., FULL for fully adjacent neighbors).

2. To view OSPF interface information:

   ```
   show ip ospf interface
   ```

   This command shows OSPF-related interface details, such as area, cost, state, and neighbors seen on the interface.

3. To see the OSPF routing table:

   ```
   show ip route ospf
   ```

   This command lists the routes learned via OSPF.

4. To check the details of the OSPF process and its database:

   ```
   show ip ospf database
   ```

   This command gives detailed information about the OSPF link-state database.

Remember that OSPF requires all areas to connect to the backbone area (area 0). If you are configuring a single area OSPF network, it should be area 0. Ensure that all participating OSPF routers have interfaces in the same area and that they are connected properly to form adjacencies. Also, router IDs should be unique across the OSPF domain. Adjust the example IP addresses, interface names, and OSPF process IDs to fit the specifics of your network.

3.4.b Point-to-point

Configuring a point-to-point OSPFv2 connection involves setting up OSPF on the interfaces that connect directly to each other. Here's how to do it:

1. Connect to your router using a console cable and open your terminal program.

2. Enter the user exec mode by typing `enable` and press `Enter`.

3. Enter the global configuration mode by typing `configure terminal` and press `Enter`.

4. Enable the OSPF process and assign a process ID by typing `router ospf [process_id]`, where `[process_id]` is a number you assign to uniquely identify the OSPF process on the router. The range is from 1 to 65535.

5. (Optional) Configure the router ID, which helps identify the router in the OSPF network, by typing `router-id [router_id]`, where `[router_id]` is the ID you want to assign to the router. If not configured manually, the router will use the highest IP address of its active interfaces.

6. Go into the interface configuration mode for the interface you want to configure by typing `interface [interface_id]`, where `[interface_id]` is the ID of the interface.

7. Enable OSPF on the interface by typing `ip ospf [process_id] area [area_id]`, where `[process_id]` is the OSPF process ID you assigned earlier and `[area_id]` is the ID of the OSPF area you want the interface to belong to.

8. Set the network type to point-to-point by typing `ip ospf network point-to-point`.

9. Save your changes by typing `exit`, then `copy running-config startup-config`.

To verify your configuration:

1. In the privileged exec mode (type `enable` and press `Enter` if you're not already in this mode), type `show ip ospf interface brief`. This command will display brief information about the OSPF interfaces, including the network type.

Please replace `[process_id]`, `[interface_id]`, `[area_id]`, and `[router_id]` with your specific OSPF parameters.

Sample Configuration for single area OSPFv2 with Point-to-point:

Configuring single area OSPFv2 for a point-to-point connection is similar to configuring OSPF for any network type, but with a point-to-point connection, there are some optimizations and configurations that can be made. Here's how to configure OSPF for a point-to-point link:

1. Access your Cisco router and enter global configuration mode:

   ```
   enable
   configure terminal
   ```

2. Enable OSPF using a process ID (the process ID is locally significant and can be any positive integer):

   ```
   router ospf 1
   ```

3. Assign a router ID (optional, but recommended for stability):

   ```
   router-id 1.1.1.1
   ```

4. Configure OSPF on the point-to-point interface:

   ```
   interface Serial0/0/0
   ip address 192.168.12.1 255.255.255.252
   ip ospf network point-to-point
   ip ospf 1 area 0
   ```

In this example, `Serial0/0/0` is the interface used for the point-to-point connection. The `ip ospf network point-to-point` command is used to explicitly specify the OSPF network type as point-to-point, which is generally the default for serial interfaces. The IP address and subnet mask will likely be different for your network.

5. (Optional) If you want all interfaces participating in OSPF to be in area 0, you can also use the network command with a wildcard mask that includes all interfaces:

   ```
   network 0.0.0.0 255.255.255.255 area 0
   ```

   However, be cautious with this command, as it will include all interfaces in OSPF, which may not be desired.

6. Save the configuration:

   ```
   write memory
   ```

Verification of OSPF Configuration on Point-to-Point Links

After configuring OSPF, you can verify the point-to-point link with the following commands:

1. To verify OSPF neighbor relationships:

   ```
   show ip ospf neighbor
   ```

   This command displays information about OSPF neighbors. On point-to-point links, the neighbor state should quickly reach the FULL state.

2. To view OSPF interface information:

   ```
   show ip ospf interface
   ```

   This command shows OSPF-related interface details, including the network type, area, timers, and neighbor count.

3. To see the OSPF routing table:

   ```
   show ip route ospf
   ```

   This command lists the routes learned via OSPF, including those from your point-to-point link.

4. To check the details of the OSPF process and its database:

   ```
   show ip ospf database
   ```

   This command gives detailed information about the OSPF link-state database.

Ensure that both ends of the point-to-point link are configured in the same OSPF area and that the OSPF network type is set correctly. The router IDs should also be unique across the OSPF domain. Adjust the example IP addresses, interface names, and OSPF process IDs to fit the specifics of your network.

3.4.c Broadcast (DR/BDR selection)

In a broadcast multi-access network, OSPF elects one router to be a Designated Router (DR) and another to be a Backup Designated Router (BDR) to manage the OSPF information exchange between routers. Here's how to configure it:

1. Connect to your router using a console cable and open your terminal program.

2. Enter the user exec mode by typing `enable` and press `Enter`.

3. Enter the global configuration mode by typing `configure terminal` and press `Enter`.

4. Enable the OSPF process and assign a process ID by typing `router ospf [process_id]`, where `[process_id]` is a number you assign to uniquely identify the OSPF process on the router. The range is from 1 to 65535.

5. (Optional) Configure the router ID, which helps identify the router in the OSPF network, by typing `router-id [router_id]`, where `[router_id]` is the ID you want to assign to the router. If not configured manually, the router will use the highest IP address of its active interfaces.

6. Go into the interface configuration mode for the interface you want to configure by typing `interface [interface_id]`, where `[interface_id]` is the ID of the interface.

7. Enable OSPF on the interface by typing `ip ospf [process_id] area [area_id]`, where `[process_id]` is the OSPF process ID you assigned earlier and `[area_id]` is the ID of the OSPF area you want the interface to belong to.

8. (Optional) To influence the DR/BDR election process, you can assign a priority to the interface by typing `ip ospf priority [priority]`, where `[priority]` is a number between 0 and 255. The router with the highest priority will become the DR. If the priorities are equal, the router with the highest router ID will become the DR.

9. Save your changes by typing `exit`, then `copy running-config startup-config`.

To verify your configuration:

1. In the privileged exec mode (type `enable` and press `Enter` if you're not already in this mode), type `show ip ospf neighbor`. This command will display information about the OSPF neighbors, including the DR and BDR.

Please replace `[process_id]`, `[interface_id]`, `[area_id]`, `[router_id]`, and `[priority]` with your specific OSPF parameters.

Sample Configuration for single area OSPFv2 with Broadcast (DR/BDR selection):

In a single-area OSPFv2 broadcast network, the OSPF protocol automatically elects a Designated Router (DR) and a Backup Designated Router (BDR) on multi-access networks like Ethernet. The DR and BDR facilitate the exchange of OSPF information by reducing the amount of LSA flooding. Here's how to configure OSPF for a broadcast network with DR/BDR election:

Sample Configuration for Single-Area OSPFv2 on a Broadcast Network

1. Access your Cisco router and enter global configuration mode:

   ```
   enable
   configure terminal
   ```

2. Enable OSPF using a process ID:

```
router ospf 1
```

3. Assign a router ID (optional, but recommended for stability and must be unique):

   ```
   router-id [router-id]
   ```

   Example:

   ```
   router-id 2.2.2.2
   ```

4. Configure OSPF on the interfaces that will participate in the OSPF process. For example, on a GigabitEthernet interface that connects to a broadcast network:

   ```
   interface GigabitEthernet0/0
   ip address 192.168.1.1 255.255.255.0
   ip ospf 1 area 0
   ```

   The above command assigns the interface to OSPF area 0. By default, OSPF treats Ethernet interfaces as broadcast and will participate in DR/BDR elections.

5. (Optional) If you wish to influence the DR/BDR election process, you can change the OSPF interface priority (the higher the priority, the more likely the router will become DR or BDR; the default priority is 1):

   ```
   interface GigabitEthernet0/0
   ip ospf priority [priority-value]
   ```

   Example:

   ```
   ip ospf priority 100
   ```

   Setting the priority to 0 will ensure the router will not be elected as a DR or BDR.

6. Save the configuration:

   ```
   write memory
   ```

Verification of OSPF Configuration on Broadcast Networks

After configuring OSPF, verify the broadcast network configuration and DR/BDR election with the following commands:

1. To verify OSPF neighbor relationships and see the elected DR and BDR:

   ```
   show ip ospf neighbor
   ```

   This command displays OSPF neighbor information, including the state, address, and role (DR, BDR, or DROTHER).

2. To view OSPF interface information and check the OSPF network type and interface priority:

```
   show ip ospf interface
   ```

   This command shows OSPF-related interface details, including the network type (broadcast by default on Ethernet interfaces), area, timers, priority, and neighbor count.

3. To see the OSPF routing table:

   ```
   show ip route ospf
   ```

   This command lists the routes learned via OSPF.

4. To check the OSPF database and ensure that LSA flooding is occurring properly:

   ```
   show ip ospf database
   ```

   This command gives detailed information about the OSPF link-state database.

Ensure that all routers on the broadcast network have OSPF configured and are in the same area. Router IDs should be unique. The OSPF network type for Ethernet interfaces defaults to broadcast, so you typically don't need to manually configure the network type unless it's been changed. Adjust the IP addresses, interface names, and OSPF process IDs to match your network configuration.

3.4.d Router ID

To configure and verify the OSPFv2 Router ID, follow the steps below:

1. Connect to your router using a console cable and launch your terminal program.
2. Enter the user exec mode by typing `enable` in the command line interface (CLI).
3. Enter the global configuration mode by typing `configure terminal`.
4. Enable the OSPF process and assign a process ID by typing `router ospf [process_id]`, where `[process_id]` is a number you assign to uniquely identify the OSPF process on the router. The range is from 1 to 65535.
5. Configure the router ID, which helps identify the router in the OSPF network, by typing `router-id [router_id]`, where `[router_id]` is the ID you want to assign to the router.
6. Save your changes by typing `exit`, then `copy running-config startup-config`.

To verify your configuration:

1. In the privileged exec mode (type `enable` and press `Enter` if you're not already in this mode), type `show ip ospf`. This command will display information about the OSPF process, including the Router ID.

Please replace `[process_id]` and `[router_id]` with your specific OSPF parameters. The router ID should be a unique, routable IP address in your network. It's often a good practice to use a loopback interface IP for the router ID, as it's not associated with a physical interface that could go down.

Sample Configuration for single area OSPFv2 with Router ID:

The Router ID (RID) is an important concept in OSPFv2. It uniquely identifies a router within an OSPF domain and is chosen based on the highest IP address of any of the router's active interfaces or can be manually set to any 32-bit value, typically formatted as an IPv4 address for ease of readability. If you configure the Router ID manually, it overrides any automatically chosen ID. Here's how to configure OSPF for a single area with a specified Router ID:

Sample Configuration for Single-Area OSPFv2 with Router ID

1. Access your Cisco router and enter global configuration mode:

   ```
   enable
   configure terminal
   ```

2. Start the OSPF process with a process ID that is locally significant:

   ```
   router ospf 1
   ```

3. Manually set the Router ID:

   ```
   router-id [Router-ID]
   ```

   Example:

   ```
   router-id 1.1.1.1
   ```

   This command sets the OSPF Router ID to 1.1.1.1. Choose an ID that is unique among all routers in the OSPF area.

4. Assign networks to the OSPF process using the `network` command. The networks specified in this command will participate in OSPF:

   ```
   network [Network-Address] [Wildcard-Mask] area [Area-ID]
   ```

   Example:

   ```
   network 192.168.1.0 0.0.0.255 area 0
   network 10.0.0.0 0.255.255.255 area 0
   ```

   These commands tell the router that any interfaces with IP addresses in the 192.168.1.0/24 and 10.0.0.0/8 networks will participate in OSPF area 0.

5. Exit OSPF configuration mode and return to global configuration mode:

   ```
   end
   ```

6. Save the configuration:

   ```
   write memory
   ```

   or

```
copy running-config startup-config
```

Note: Manually setting a Router ID is particularly useful if your router currently does not have any active interfaces or if you want to ensure a consistent Router ID regardless of interface status or IP address changes.

Verification of OSPF Configuration and Router ID

After configuring OSPF with a specific Router ID, you can verify the configuration with the following commands:

1. To verify the OSPF Router ID and other OSPF settings:

   ```
   show ip ospf
   ```

   This command displays OSPF general information, including the Router ID, OSPF process ID, and other configuration settings.

2. To see the OSPF neighbors and ensure they are recognizing the Router ID correctly:

   ```
   show ip ospf neighbor
   ```

   This command lists OSPF neighbors, their states, and their associated Router IDs.

3. To view OSPF-related information on the interfaces:

   ```
   show ip ospf interface
   ```

   This command shows OSPF-related interface details, including the OSPF area, cost, priority, and other interface-specific OSPF settings.

It's important to configure OSPF Router IDs before any OSPF neighbor relationships have been established. If you change a Router ID after OSPF has been established, you will need to restart the OSPF process for the change to take effect, which can be done by either using the `clear ip ospf process` command or by reloading the router. Remember, the Router ID must be unique within the entire OSPF domain.

4.1 Configure and verify inside source NAT using static and pools

Network Address Translation (NAT) allows a device like a router to translate IP addresses from one network to another. This is often used to allow multiple devices on a local network to share a single public IP address. Here's how to configure inside source NAT using static mapping and pools on a Cisco router:

For Static NAT:

1. Connect to your router using a console cable and launch your terminal program.

2. Enter the user exec mode by typing `enable` and press `Enter`.

3. Enter the global configuration mode by typing `configure terminal` and press `Enter`.

4. Configure a static NAT translation by typing `ip nat inside source static [local_IP] [global_IP]`, where `[local_IP]` is the IP address of the host on the local network and `[global_IP]` is the IP address that the host will use on the global network.

5. Press `Enter`.

6. Go into the interface configuration mode for the inside interface (connected to the local network) by typing `interface [inside_interface_id]`, where `[inside_interface_id]` is the ID of the inside interface.
7. Enable NAT on the inside interface by typing `ip nat inside`.
8. Press `Enter`.
9. Go into the interface configuration mode for the outside interface (connected to the global network) by typing `interface [outside_interface_id]`, where `[outside_interface_id]` is the ID of the outside interface.
10. Enable NAT on the outside interface by typing `ip nat outside`.
11. Press `Enter`.
12. Save your changes by typing `exit`, then `copy running-config startup-config` and press `Enter`.

Sample configuration for the Static NAT on Cisco device:

Static Network Address Translation (NAT) on a Cisco device translates a single unregistered (inside, private) IP address to a single registered (outside, public) IP address, and vice versa. This is often used to allow an internal device, such as a web server, to be reachable from the outside network while using a private IP address internally.

Here is a sample configuration for Static NAT on a Cisco router:

1. Access your Cisco router and enter global configuration mode:

```
enable
configure terminal
```

2. Define the inside and outside interfaces for NAT. Typically, the inside interface is connected to the private network, and the outside interface is connected to the public network:

```
interface GigabitEthernet0/0
ip nat inside
exit

interface GigabitEthernet0/1
ip nat outside
exit
```

Replace `GigabitEthernet0/0` and `GigabitEthernet0/1` with the actual interface names on your device.

3. Create an access control list (ACL) to specify the local (inside) IP address that will be translated:

```
access-list 1 permit 192.168.1.10
```

In this example, `192.168.1.10` represents the inside IP address of the device you want to enable for Static NAT.

4. Establish the static NAT translation:

```
ip nat inside source static 192.168.1.10 [Public-IP-Address]
```

Replace `[Public-IP-Address]` with the public IP address that you want to map to the inside IP address. For example:

```
ip nat inside source static 192.168.1.10 203.0.113.10
```

This command sets up a static translation where traffic that reaches the router's outside interface with the destination IP address of `203.0.113.10` will be translated to `192.168.1.10`.

5. Save the configuration:

```
write memory
```

Verification of Static NAT Configuration

To verify that the static NAT is configured correctly, you can use the following command:

```
show ip nat translations
```

This command displays the NAT translation table, including static entries. You should see an entry that maps the inside local address to the inside global address.

If you've just applied the configuration and the translation is not appearing, you can try to generate traffic from the inside host to the outside to trigger the NAT and then check again.

Remember that Static NAT will allow inbound connections initiated from the outside to the inside private address if the outside-to-inside translation is specified. You may need to configure additional security measures such as access lists or firewall rules to protect your network. Static NAT can also be used in conjunction with Port Address Translation (PAT) to map a single public IP address to multiple private IP addresses distinguished by different port numbers.

For Dynamic NAT with a Pool:

1. Follow steps 1-3 from the static NAT configuration above.

2. Define a pool of global addresses for NAT to use by typing `ip nat pool [pool_name] [start_IP] [end_IP] netmask [subnet_mask]`, where `[pool_name]` is the name you want to give to the address pool, `[start_IP]` and `[end_IP]` define the range of addresses in the pool, and `[subnet_mask]` is the subnet mask of the global network.

3. Press `Enter`.

4. Define which local addresses should be translated by creating an access list. For example, to translate all addresses in the local network, type `access-list 1 permit [local_network] [wildcard_mask]`, where `[local_network]` is the network address of the local network and `[wildcard_mask]` is the inverse of the subnet mask.

5. Press `Enter`.

6. Configure NAT to use the pool and access list by typing `ip nat inside source list 1 pool [pool_name]`, where `[pool_name]` is the name of the address pool you created earlier.

7. Press `Enter`.

8. Follow steps 6-12 from the static NAT configuration above.

To verify your NAT configuration, type `show ip nat translations` in the privileged exec mode to display the current NAT translations.

Please replace `[local_IP]`, `[global_IP]`, `[inside_interface_id]`, `[outside_interface_id]`, `[pool_name]`, `[start_IP]`, `[end_IP]`, `[subnet_mask]`, `[local_network]`, and `[wildcard_mask]` with your specific network parameters.

Sample configuration for the Dynamic NAT with a Pool on Cisco device:

Dynamic Network Address Translation (NAT) with a pool of addresses allows you to translate inside local addresses to a pool of global addresses. This is useful when you have a range of public IP addresses available and want to allow multiple devices on a private network to access external networks such as the internet.

Here's how to configure dynamic NAT with a pool on a Cisco router:

1. Access your Cisco router and enter global configuration mode:

   ```
   enable
   configure terminal
   ```

2. Define the pool of public IP addresses that will be used for translation:

   ```
   ip nat pool [pool-name] [start-ip] [end-ip] netmask [subnet-mask]
   ```

   Example:

   ```
   ip nat pool NAT-POOL 203.0.113.10 203.0.113.20 netmask 255.255.255.0
   ```

   This command creates a NAT pool named `NAT-POOL` with a range of public IP addresses from `203.0.113.10` to `203.0.113.20` and a subnet mask of `255.255.255.0`.

3. Create an access control list (ACL) to match the local (inside) IP addresses that are allowed to be translated:

   ```
   access-list [acl-number] permit [inside-network] [wildcard-mask]
   ```

   Example:

   ```
   access-list 10 permit 192.168.1.0 0.0.0.255
   ```

   This ACL permits any device on the `192.168.1.0/24` network to be translated using NAT.

4. Link the ACL and the pool together with a NAT statement:

   ```
   ip nat inside source list [acl-number] pool [pool-name]
   ```

   Example:

   ```
   ip nat inside source list 10 pool NAT-POOL
   ```

This command tells the router to translate any inside local addresses that match ACL 10 to a global address from the `NAT-POOL`.

5. Identify which interfaces are inside and which are outside relative to the NAT translation:

    ```
    interface [inside-interface-name]
    ip nat inside
    exit

    interface [outside-interface-name]
    ip nat outside
    exit
    ```

    Example:

    ```
    interface GigabitEthernet0/0
    ip nat inside
    exit

    interface GigabitEthernet0/1
    ip nat outside
    exit
    ```

    Replace `[inside-interface-name]` and `[outside-interface-name]` with the actual interface names on your router.

6. Save the configuration:

    ```
    write memory
    ```

    or

    ```
    copy running-config startup-config
    ```

Verification of Dynamic NAT Configuration

After configuring dynamic NAT with a pool, verify that it's working correctly:

1. Check the NAT translation table:

    ```
    show ip nat translations
    ```

    This command shows active NAT translations. After the hosts start sending traffic, you should see their inside local addresses being translated to inside global addresses from the pool.

2. View the NAT statistics and verify that the translations are working as expected:

    ```
    show ip nat statistics
    ```

    This command provides an overview of the NAT configuration, including the total number of active translations.

Remember to configure your device interfaces with the correct IP addressing and ensure they are operational. Also, make sure that the devices on the inside network use the router as their default gateway, or that the router is included in the path of the devices' traffic to the outside network.

4.2 Configure and verify NTP operating in a client and server mode

Configuring and verifying Network Time Protocol (NTP) operating in a client-server mode mainly involves setting up one device as a server and another as a client, then ensuring that the client can synchronize its time with the server.

Here's a simplified version of how you can do this on a Cisco device:

1. **Configuring the NTP Server:**

On your server device, enter the following commands in configuration mode:

```
Router> enable
Router# configure terminal
Router(config)# ntp master 1
Router(config)# end
```

This sets the device as an NTP server with a stratum level of 1, which is the highest level of accuracy.

2. **Configuring the NTP Client:**

On your client device, you would point it towards the server using the server's IP address:

```
Router> enable
Router# configure terminal
Router(config)# ntp server 192.168.1.1
Router(config)# end
```

Replace "192.168.1.1" with the IP address of your server.

3. **Verifying NTP Operation:**

You can verify that NTP is functioning correctly on both the client and server with the command:

```
Router> show ntp status
```

On the server, it should show `synchronized to local sys`, and on the client, it should show `synchronized to 192.168.1.1`, or whatever the IP of your server is.

Keep in mind that NTP synchronization may take several minutes. Also, the specifics of these instructions can vary depending on the exact equipment and software you're using. Always refer to your device's specific documentation for the most accurate information.

4.6 Configure and verify DHCP client and relay

Configuring and verifying a DHCP client and relay involves setting up a device to receive an IP address from a DHCP server and a relay to forward DHCP requests between networks.

Here's a simplified version of how you can do this on a Cisco device:

1. **Configuring the DHCP Client:**

On your client device, enter the following commands in configuration mode:

```
Router> enable
Router# configure terminal
Router(config)# interface gigabitEthernet 0/0
Router(config-if)# ip address dhcp
Router(config-if)# no shutdown
Router(config-if)# end
```

This sets the device's GigabitEthernet 0/0 interface to automatically receive an IP address from a DHCP server.

2. **Configuring the DHCP Relay:**

On your relay device, you would set it to forward DHCP requests to a specific server:

```
Router> enable
Router# configure terminal
Router(config)# interface gigabitEthernet 0/0
Router(config-if)# ip helper-address 192.168.1.1
Router(config-if)# end
```

Replace "192.168.1.1" with the IP address of your DHCP server. This will forward all DHCP requests that the device receives on its GigabitEthernet 0/0 interface to the specified server.

3. **Verifying DHCP Operation:**

You can verify that the DHCP client and relay are functioning correctly with the commands:

```
Router> show ip interface brief
```

For the client, this will show the IP address that it has received from the DHCP server.

```
Router> show ip helper-address
```

For the relay, this will show the IP address of the DHCP server that it is forwarding requests to.

Keep in mind that DHCP may take a few moments to assign an IP address. Also, the specifics of these instructions can vary depending on the exact equipment and software you're using. Always refer to your device's specific documentation for the most accurate information.

4.8 Configure network devices for remote access using SSH

Secure Shell (SSH) is a cryptographic network protocol for securely accessing network devices over an unsecured network. Here's a basic procedure on how to configure a Cisco device for remote access using SSH:

1. **Configure Hostname and Domain Name:**

SSH requires a hostname and domain name to generate necessary RSA keys. If your router doesn't have them configured yet, you can do so as follows:

```
Router> enable
Router# configure terminal
Router(config)# hostname MyRouter
MyRouter(config)# ip domain-name mydomain.com
```

2. **Generate RSA Key:**

SSH works by using a pair of keys: one public and one private. You need to generate these keys on your router:

```
MyRouter(config)# crypto key generate rsa
```

When prompted, choose a modulus of at least 1024 bits for security.

3. **Configure User and Password:**

You need to configure the local database with usernames and passwords to use for SSH login:

```
MyRouter(config)# username admin password mypassword
```

4. **Enable SSH on the VTY lines:**

Now enable SSH access on the VTY lines and specify the local user database for authentication:

```
MyRouter(config)# line vty 0 4
MyRouter(config-line)# transport input ssh
MyRouter(config-line)# login local
MyRouter(config-line)# exit
```

5. **Set SSH Version:**

To increase security, it is recommended to set the SSH version to 2:

```
MyRouter(config)# ip ssh version 2
```

6. **Verify SSH Configuration:**

You can verify SSH access to your router by using the following command:

```
MyRouter# show ip ssh
```

This will display the SSH version and RSA key pair details.

The device is now configured for SSH remote access. You can SSH into the router using the created user credentials. Always make sure to replace "MyRouter", "mydomain.com", "admin", and "mypassword" with your own specific values. Also, these instructions may vary depending on your specific equipment and software. Always refer to your device's documentation for the most accurate information.

5.3 Configure and verify device access control using local passwords

To configure and verify device access control using local passwords on a Cisco device, you would generally follow these steps:

1. **Setting Up a Password for Privileged Mode (Enable Password):**

```
Router> enable
Router# configure terminal
Router(config)# enable secret mypassword
```

Here, replace “mypassword” with the password you want to use for privileged mode.

2. **Setting Up a Password for Console Access:**

```
Router(config)# line console 0
Router(config-line)# password mypassword
Router(config-line)# login
Router(config-line)# exit
```

Again, replace “mypassword” with the password you want to use for console access.

3. **Setting Up a Password for Remote (VTY) Access:**

```
Router(config)# line vty 0 4
Router(config-line)# password mypassword
Router(config-line)# login
Router(config-line)# exit
Router(config)# exit
```

Replace “mypassword” with the password you want to use for remote access.

4. **Verifying Access Control:**

To verify that the passwords are working, you can simply try accessing the device through the console or remotely via Telnet or SSH (if configured). You should be prompted for the password.

Remember to replace “mypassword” with the password you want to use. The `enable secret` command provides a higher security level for protecting privileged access than the older `enable password` command. Also, always refer to your device’s specific documentation for the most accurate information.

5.6 Configure and verify access control lists

Access Control Lists (ACLs) are used to filter network traffic on Cisco routers. Here’s a simplified version of how you can configure and verify standard ACLs:

1. **Configuring an Access Control List:**

You can configure a standard ACL to permit or deny traffic from a certain IP address. For example, to deny traffic from the IP address 192.168.1.1, you would enter the following commands:

```
Router> enable
Router# configure terminal
Router(config)# access-list 1 deny 192.168.1.1 0.0.0.0
Router(config)# access-list 1 permit any
Router(config)# end
```

The first command denies traffic from 192.168.1.1, and the second permits all other traffic. The number “1” is the ACL number.

2. **Applying the ACL to an Interface:**

Next, apply the ACL to an interface, for example, the GigabitEthernet 0/0 interface:

```
Router> enable
Router# configure terminal
Router(config)# interface gigabitEthernet 0/0
Router(config-if)# ip access-group 1 in
Router(config-if)# end
```

This applies the ACL to incoming traffic on the GigabitEthernet 0/0 interface.

3. **Verifying the ACL:**

You can view the configuration of your ACLs with the command:

```
Router# show access-lists
```

This will display all ACLs configured on the router, including their permit and deny statements.

Remember, the specifics of these instructions can vary depending on the exact equipment and software you’re using. Always refer to your device’s specific documentation for the most accurate information. Also, be careful when configuring ACLs, as incorrect configurations can cause network connectivity issues.

5.7 Configure Layer 2 security features (DHCP snooping, dynamic ARP inspection, and port security)

Configuring Layer 2 security features on a Cisco switch involves setting up DHCP snooping, dynamic ARP inspection, and port security. Here’s a basic procedure for each:

1. **DHCP Snooping:**

DHCP snooping filters out untrusted DHCP messages and prevents DHCP spoofing attacks.

```
Switch> enable
Switch# configure terminal
Switch(config)# ip dhcp snooping
Switch(config)# ip dhcp snooping vlan 1
Switch(config)# interface fastEthernet 0/1
Switch(config-if)# ip dhcp snooping trust
Switch(config-if)# end
```

Replace “vlan 1” with your VLAN number and “fastEthernet 0/1” with your interface name. The trusted interface should be the one facing the DHCP server.

2. **Dynamic ARP Inspection (DAI):**

DAI validates ARP packets in a network. It prevents ARP spoofing attacks.

```
Switch> enable
Switch# configure terminal
Switch(config)# ip arp inspection vlan 1
Switch(config)# interface fastEthernet 0/1
Switch(config-if)# ip arp inspection trust
Switch(config-if)# end
```

Replace “vlan 1” with your VLAN number and “fastEthernet 0/1” with your interface name. The trusted interface should be the one facing the DHCP server.

3. **Port Security:**

Port Security allows you to restrict input to an interface by limiting the MAC addresses of the stations allowed to access the port.

```
Switch> enable
Switch# configure terminal
Switch(config)# interface fastEthernet 0/1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security maximum 1
Switch(config-if)# switchport port-security mac-address sticky
Switch(config-if)# switchport port-security violation restrict
Switch(config-if)# end
```

Replace “fastEthernet 0/1” with your interface name. This configuration allows only one MAC address to access the interface, learns the MAC address dynamically, and restricts traffic with unknown source addresses.

Remember to always refer to your device’s specific documentation for the most accurate information.

5.10 Configure WLAN using WPA2 PSK using the GUI

To set up a WLAN using WPA2 PSK via the GUI, you’ll typically need to access your wireless router or access point’s web interface and adjust the wireless security settings. Here’s a basic step-by-step guide:

1. **Access your router’s web interface:**
    - Open a web browser on a device connected to your network.
    - Type the IP address of your router into the address bar (common default IP addresses are 192.168.0.1, 192.168.1.1, or 192.168.1.254).
    - Press Enter.

2. **Log in:**
    - Enter your username and password when prompted. If you haven’t changed these, they’ll be the factory default credentials.

3. **Navigate to the Wireless Settings:**

- This option is usually located in a main menu or sidebar. The exact navigation can vary depending on your router brand and model. Look for “Wireless,” “Wireless Settings,” “Wireless Setup,” or something similar.

4. **Configure your WLAN settings:**
    - Set your SSID (network name). This is the name that will be visible when searching for available wireless networks.
    - Set the Mode to “WPA2-PSK” or “WPA2 Personal” depending on your router model.
    - Enter your preferred password into the “Pre-Shared Key” or “Password” field. This should be a strong password to help protect your network.
5. **Save your settings:**
    - Click “Save,” “Apply,” or similar to save your changes.
6. **Test your connection:**
    - Try connecting to your WiFi network using the new SSID and password from a wireless device.

Please note that the exact steps and terms may vary depending on the brand and model of your wireless router or access point. Always refer to the specific manual or online help pages if you’re unsure.