1)Try Test-Connection and nslookup commands for below websites

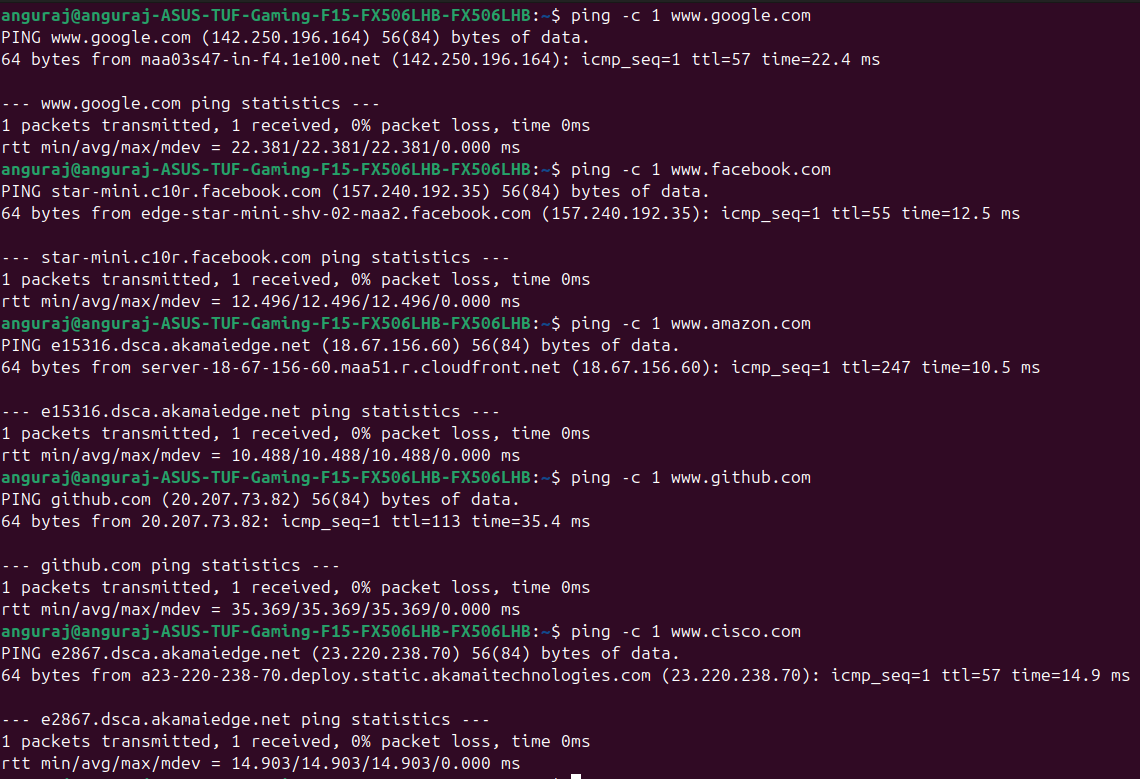
www.google.com

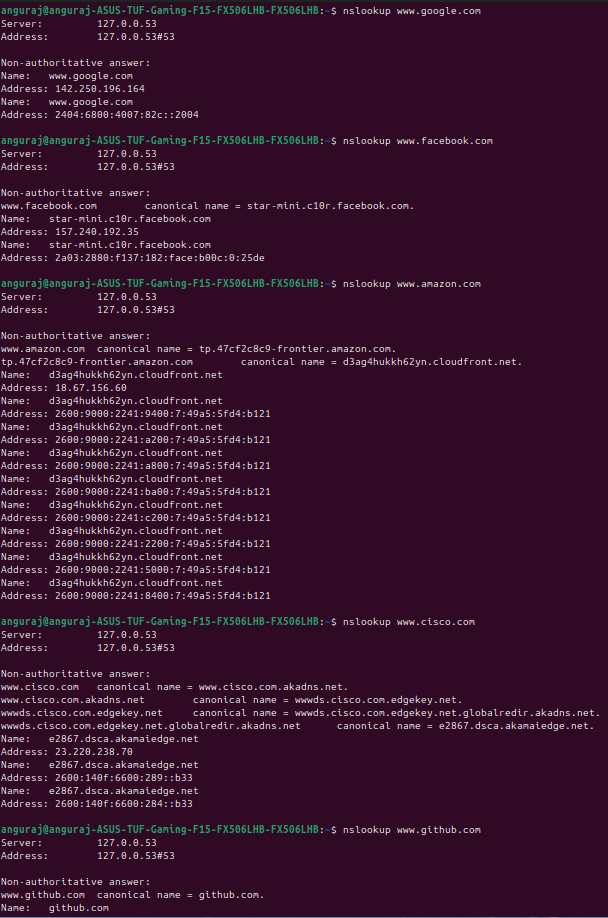
www.facebook.com

www.amazon.com

www.github.com

www.cisco.com





2.Use Wireshark to capture and analyze DNS, TCP, UDP traffic and packet header, packet flow, options and flags

**DNS**: Resolves domain names to IPs.

**TCP**: Connection-oriented, reliable communication with flags and sequence numbers.

**UDP**: Connectionless, faster but less reliable.  
  
**DNS Traffic Analysis**

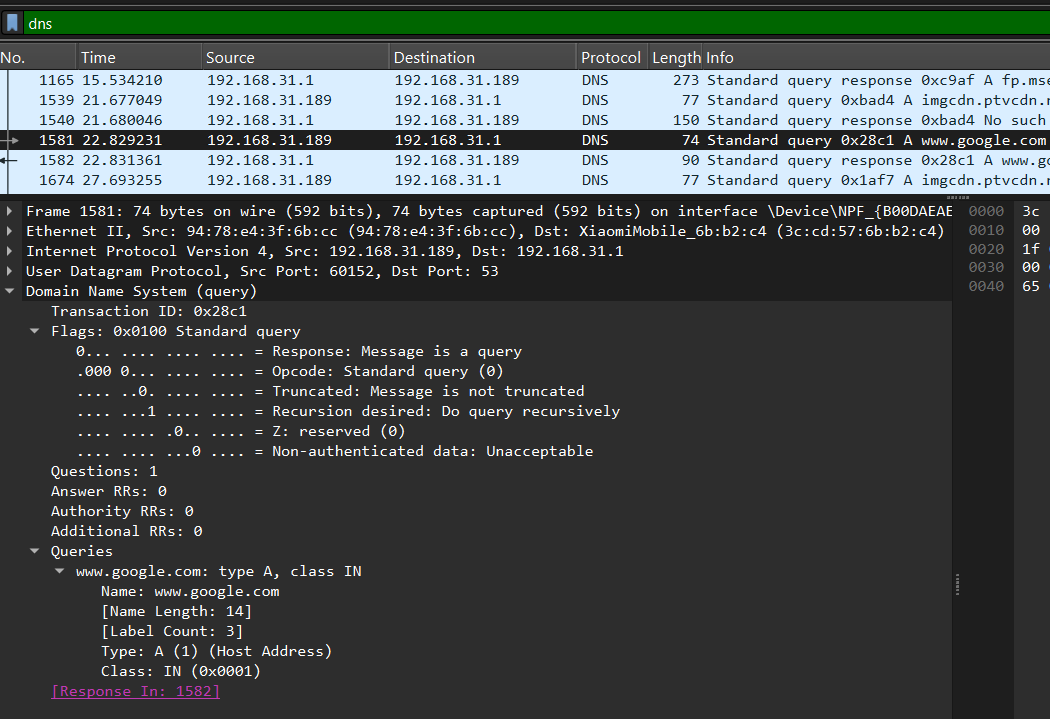
#### **Filter DNS Packets:** dns

#### **Key Fields to Analyze:**

* **Transaction ID**: Unique identifier for each query.
* **Flags**: Identifies query (0x0000) or response (0x8000).
* **Queries & Responses**: Shows domain resolution process.
* **Record Type**: A (IPv4), AAAA (IPv6), CNAME, MX.

#### **Example DNS Query Analysis:**

* A client requests www.google.com → Wireshark shows a query to a DNS server.
* The DNS server responds with an IP address.



### **TCP Traffic Analysis**

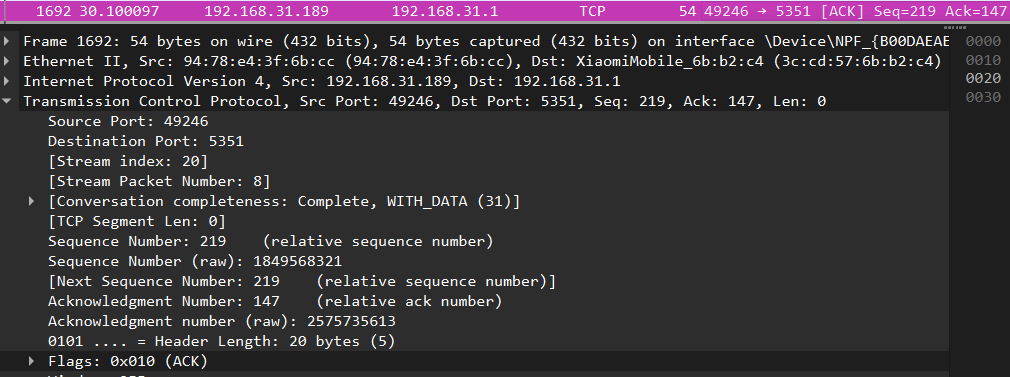
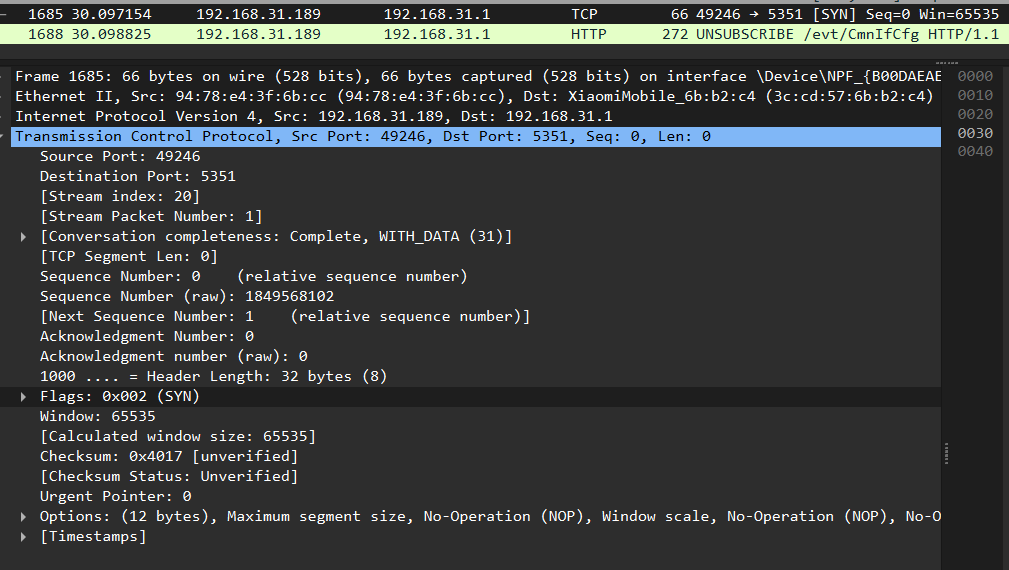
#### **Filter TCP Packets:** tcp

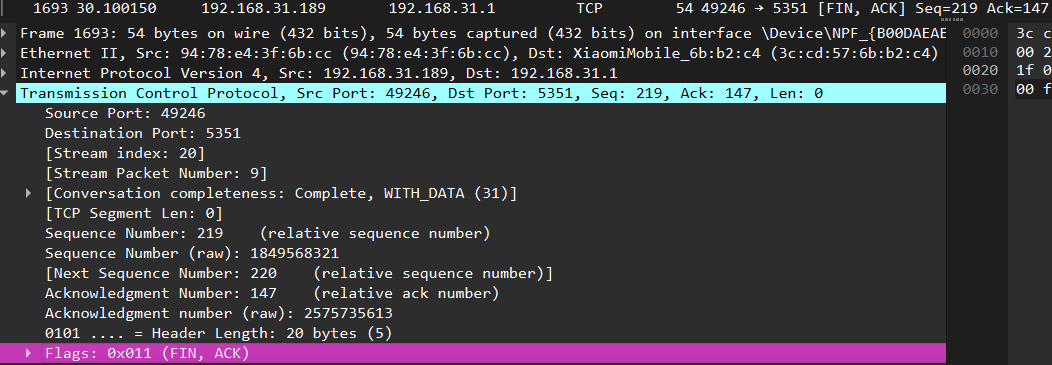
#### **Key Fields to Analyze:**

* **Source & Destination Ports**: Identifies the service (80, 443, 22).
* **Sequence & Acknowledgment Numbers**: Tracks communication flow.
* **TCP Flags:**
  + **SYN** (0x02) - Connection initiation.
  + **ACK** (0x10) - Acknowledgment.
  + **FIN** (0x01) - Connection termination.
  + **RST** (0x04) - Reset connection.

#### **Example TCP 3-Way Handshake Analysis:**

1. **SYN**: Client sends SYN to initiate connection.
2. **SYN-ACK**: Server responds with SYN-ACK.
3. **ACK**: Client sends final ACK, establishing a connection.





### **UDP Traffic Analysis**

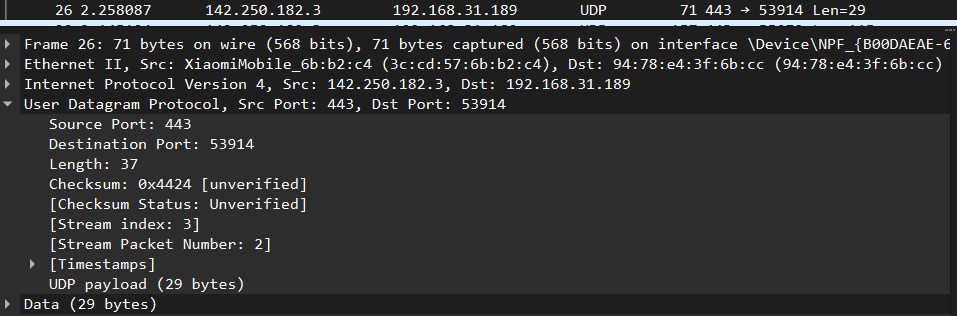
#### **Filter UDP Packets:** udp

#### **Key Fields to Analyze:**

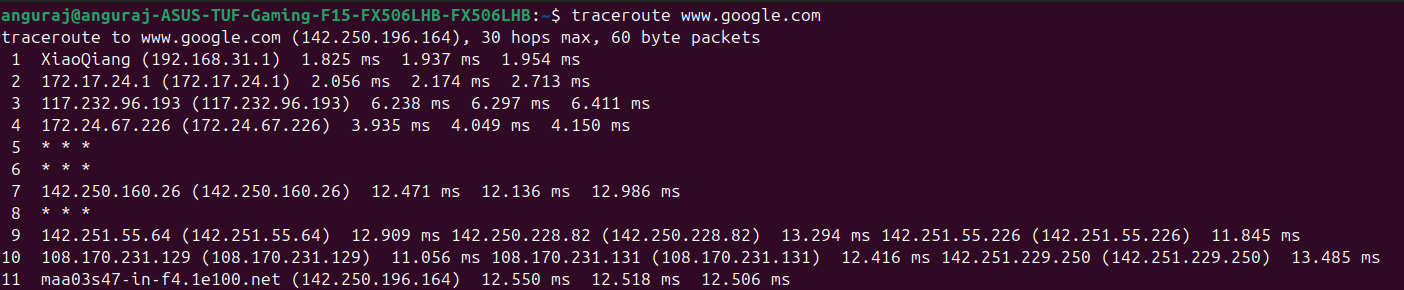
* **Source & Destination Ports**: Example - 53 (DNS), 67-68 (DHCP).
* **Length**: Size of the UDP payload.
* **Checksum**: Validates packet integrity.

#### **Example UDP Analysis:**

* DNS requests typically use UDP.



3.Explore traceroute/tracert for different websites eg:google.com and analyse the parameters in the output and explore different options for traceroute command



### **Columns Explanation**

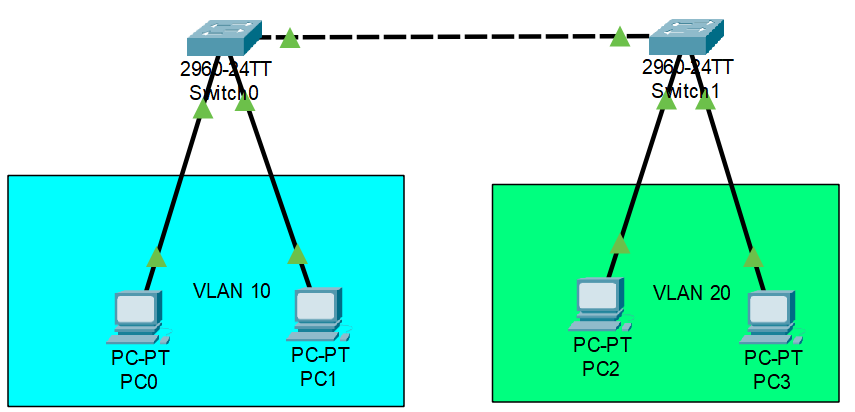
* **Hop Number**: Numbered list showing each router in the path.
* **IP Address / Hostname**: IP or resolved domain name of the router.
* **RTT (Round Trip Time) in ms**: Three response times indicate how long it took for a packet to reach the router and return.

### **Key Observations**

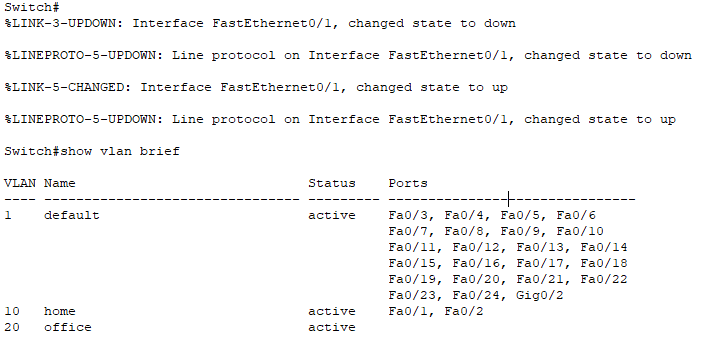
* \* \* \*: Indicates a timeout or a firewall blocking ICMP packets.
* Increasing latencies (ms values): Can indicate network congestion or distance.
* Private IPs (192.168.x.x / 10.x.x.x): Represent internal networks.

**Use Cisco packet tracer for the below**

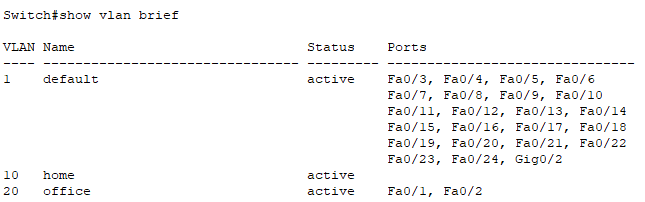
**Q4. Set up trunk ports between switches and try ping between different VLANS**

****

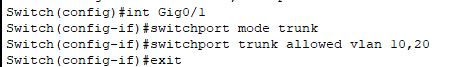
**Configuration in switch 0 (VLAN 10):**

****

**Configuration in switch 1 (VLAN 20):**

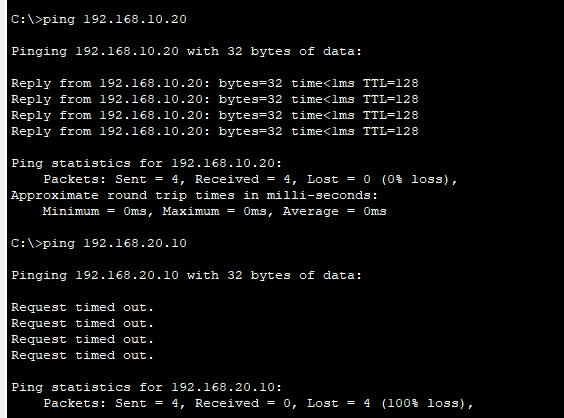


**Configuration for Trunk port between Switch 0 and switch 1 :**

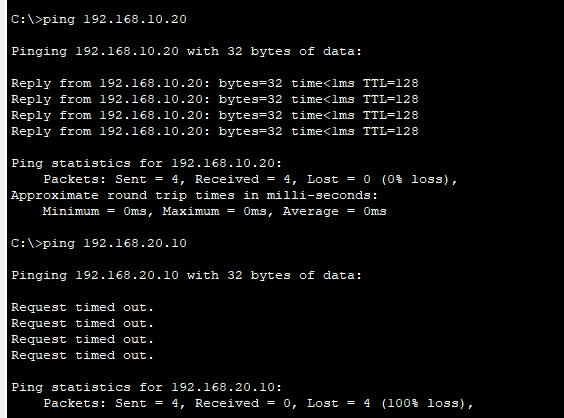
****

****

**Ping from PC0 to PC1 (same VLAN):**

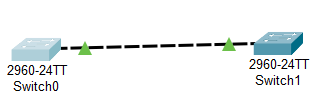
****

**Ping from PC0 to PC2 (different VLAN):**

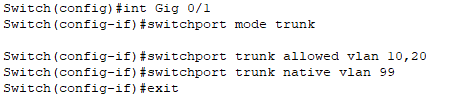
****

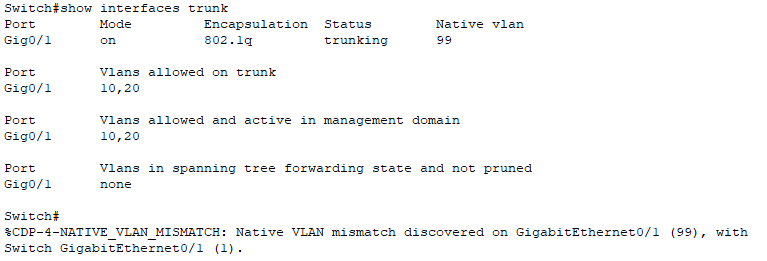
**Q5. Change the native VLAN on a trunk port. Test for VLAN mismatches and troubleshoot.**

**Two switches 0 and 1 connected with Crossover cable**

****

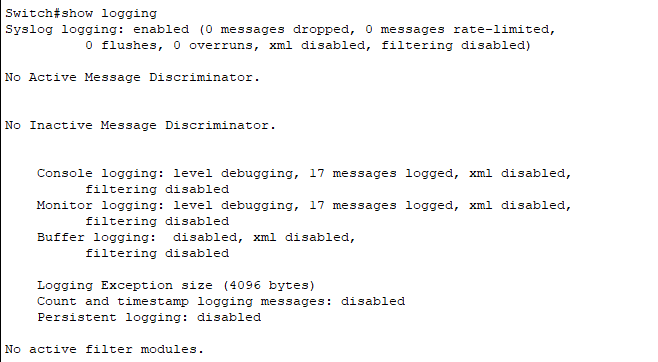
**Native VLAN configuration in switch 0:**

****

**Native VLAN mismatch issue:  
  
**

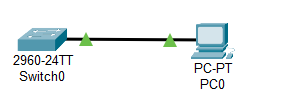
**Fix the mismatch in switch 1 :**

****

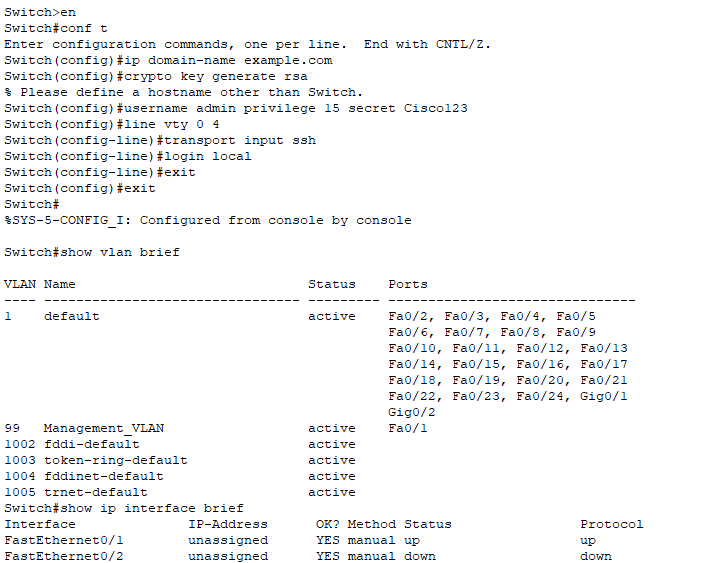
****

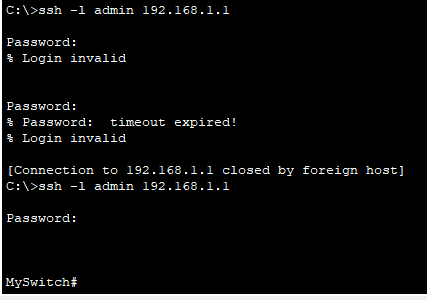
**Q6 .Configure a management VLAN and assign an IP address for remote access.**

**Test SSH or Telnet access to the switch.**

****

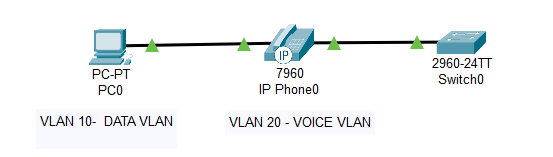
**Switch Configuration**

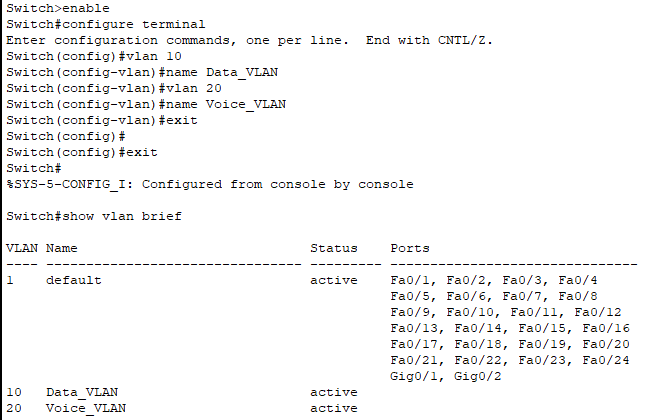
****

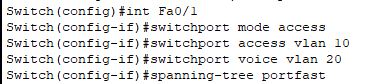
**SSH test at PC:  
  
**

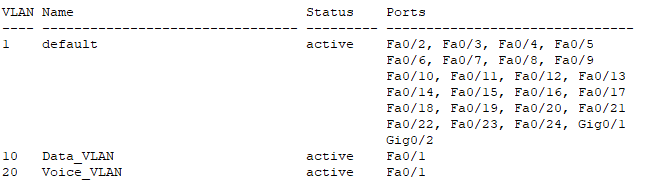
**Q7. You have a Cisco switch and a VoIP phone that needs to be placed in a voice VLAN (VLAN 20).**

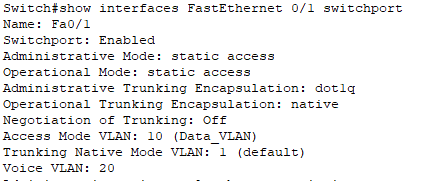
**The data for the PC should remain in a separate VLAN (VLAN 10). Configure the switch port to support both voice and data traffic.**

****

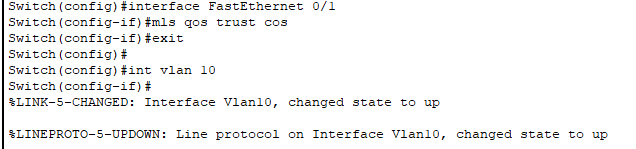
**Creating VLANs  
**

****

****

****

**Enable QoS ( for Voice Traffic)**

****

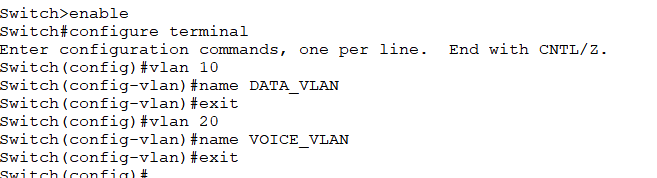
**Q8. You configured VLANs 10 and 20 on your switch and assigned ports to each VLAN. However,**

**devices in VLAN 10 cannot communicate with devices in VLAN 20. Troubleshoot the issue.**

**Q9. Try Inter VLAN routing with Router**

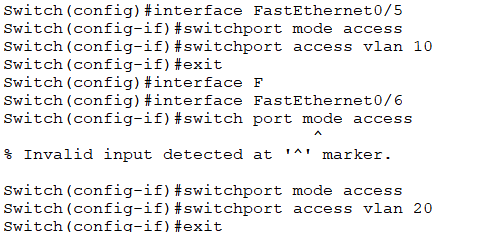
#### **1. VLAN Creation**

Ensure that VLANs 10 and 20 are properly created on the switch.



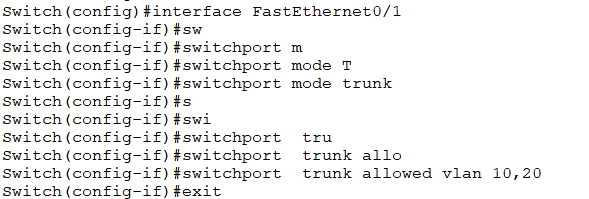
#### **2. Configuring Switch Ports**

Assign the respective switch ports to VLAN 10 and VLAN 20 as access ports.



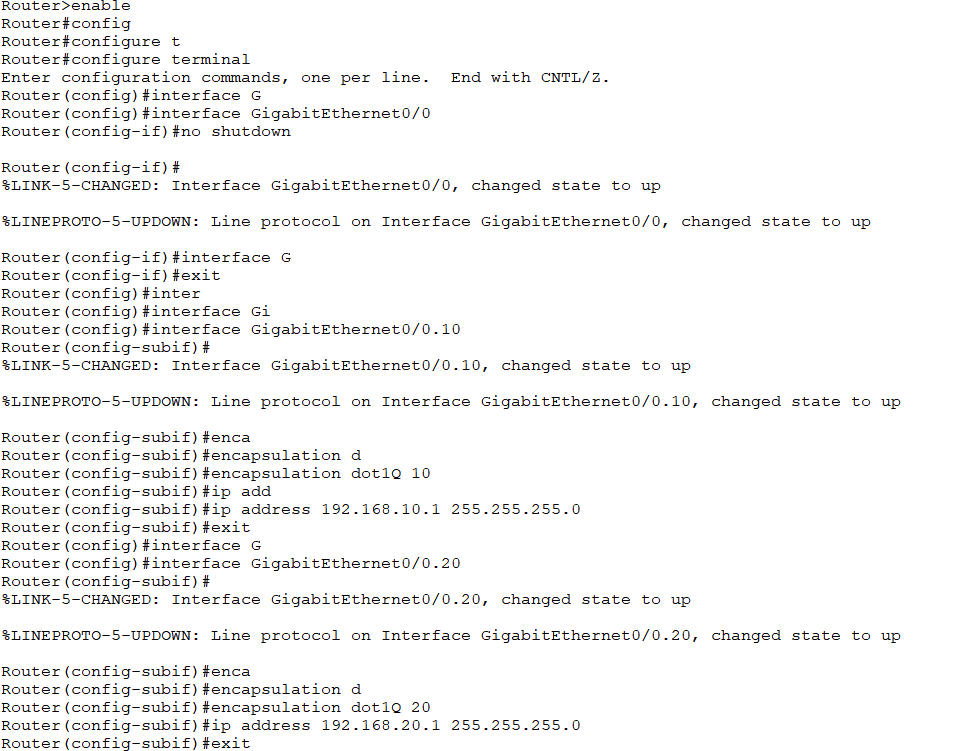
#### **3. Setting Up a Trunk Port**

Configure a trunk port to allow VLAN traffic to pass between switches and the router.



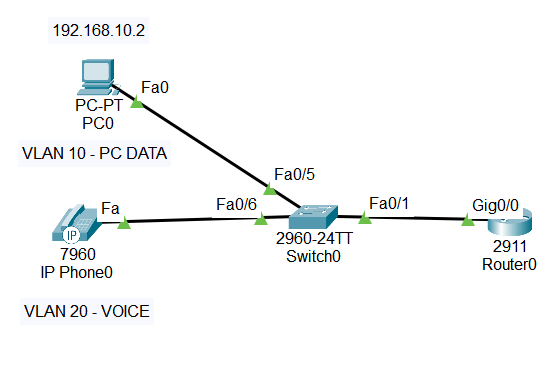
#### **4. Configuring Inter-VLAN Routing on the Router**

Since VLANs are separate broadcast domains, configure inter-VLAN routing on a router (Router-on-a-Stick) to enable communication between VLANs.



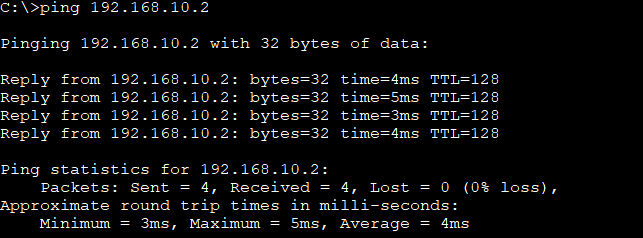
#### **5. Network Configuration in Cisco Packet Tracer**

Simulate the setup in **Cisco Packet Tracer** to verify configurations and identify potential issues.



#### **6. Ping Results and Testing**

Test connectivity between VLANs using the **ping** command to ensure proper communication.

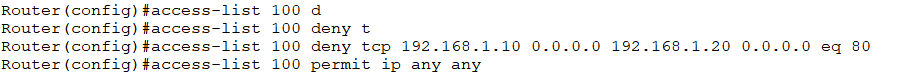
****

**Q10. Implement ACLs to restrict traffic based on source and destination ports.**

To control network traffic and restrict access based on specific ports, follow these steps:

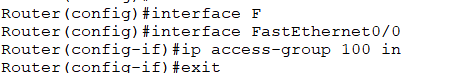
#### **1. Creating ACL on the Router**

Define an Access Control List (ACL) to restrict **PC0** from accessing **PC1** on specific ports.



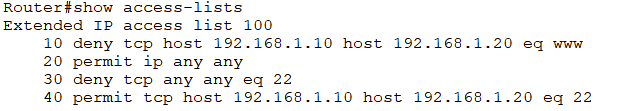
#### **2. Applying ACL to the Router's Interface**

Attach the ACL to the appropriate router interface to enforce traffic restrictions.



#### **3. Verifying ACL Rules**

Check the applied ACL configurations to ensure they are correctly implemented.



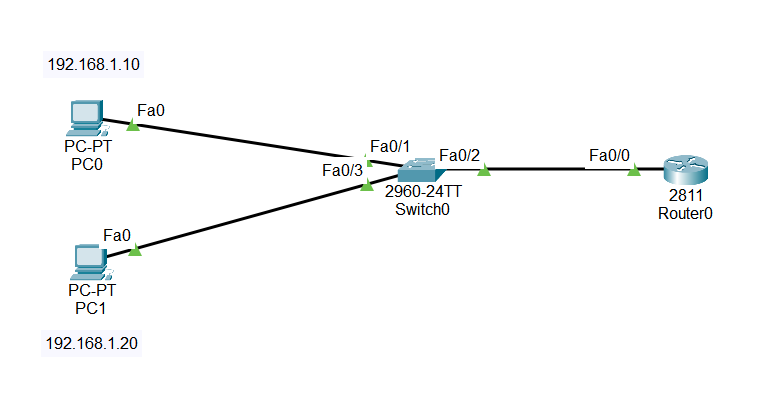
#### **4. Testing Unauthorized Traffic**

* **Simulate restricted access:** Verify that unauthorized traffic, such as SSH, is blocked.
* **Check web access:** Confirm that HTTP traffic from **PC0** is denied.

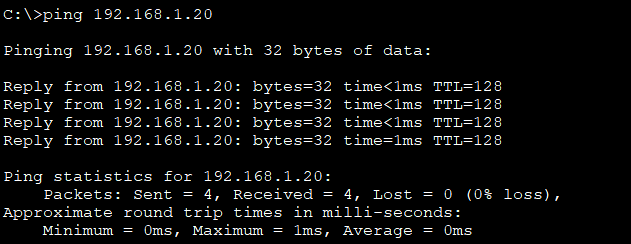


#### **5. Network Configuration in Cisco Packet Tracer**

Test ACL behavior in **Cisco Packet Tracer** to validate network restrictions.



#### **6. Validating Results**

* **Ping test:** Ensure that **ICMP (ping) works**, confirming that only HTTP is blocked.
* **Web browser test:** Check that **PC0 fails to access web pages due to ACL restrictions**.
* ****

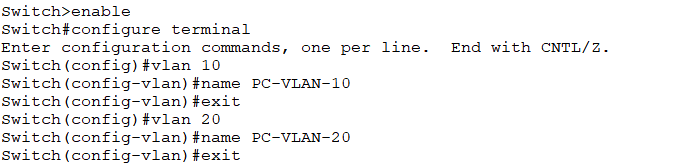
**Q11. Configure a standard Access Control List (ACL) on a router to permit traffic from a specific IP range.**

**Test connectivity to verify the ACL is working as intended.**

To control network access using a standard Access Control List (ACL), follow these steps:

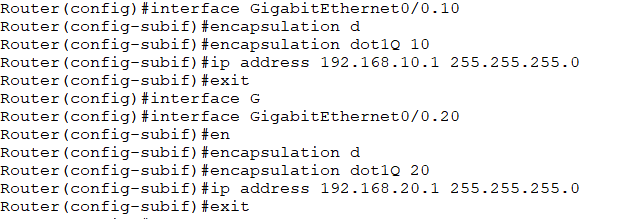
#### **1. Configuring VLANs on the Switch**

Assign VLANs to switch ports to segment the network and ensure proper traffic isolation.



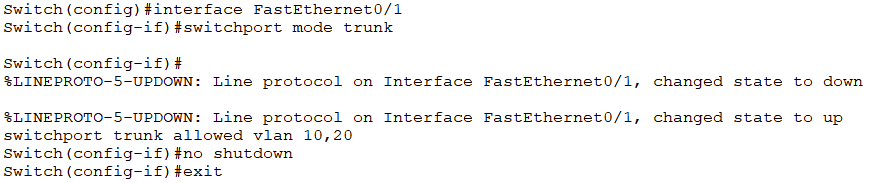
#### **2. Configuring Inter-VLAN Routing**

Enable routing between VLANs to allow communication across different network segments.



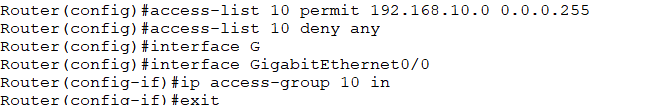
#### **3. Setting Up the Switch Trunk Port**

Configure trunk ports to allow VLAN traffic to pass between switches and the router.



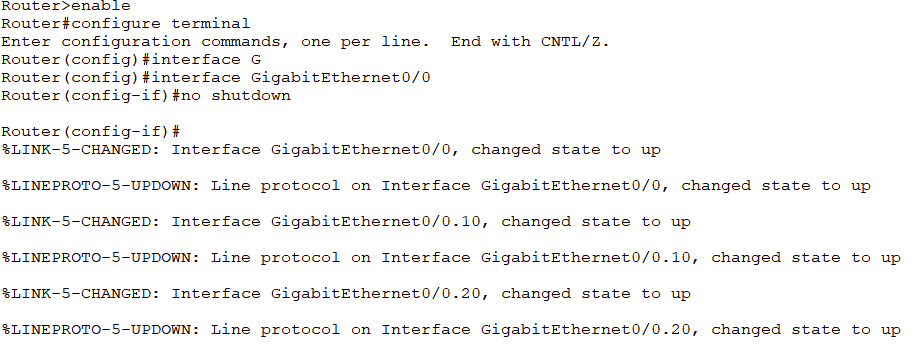
#### **4. Applying a Standard ACL on the Router**

Create and apply an ACL to permit traffic from a specific IP range while restricting other access.



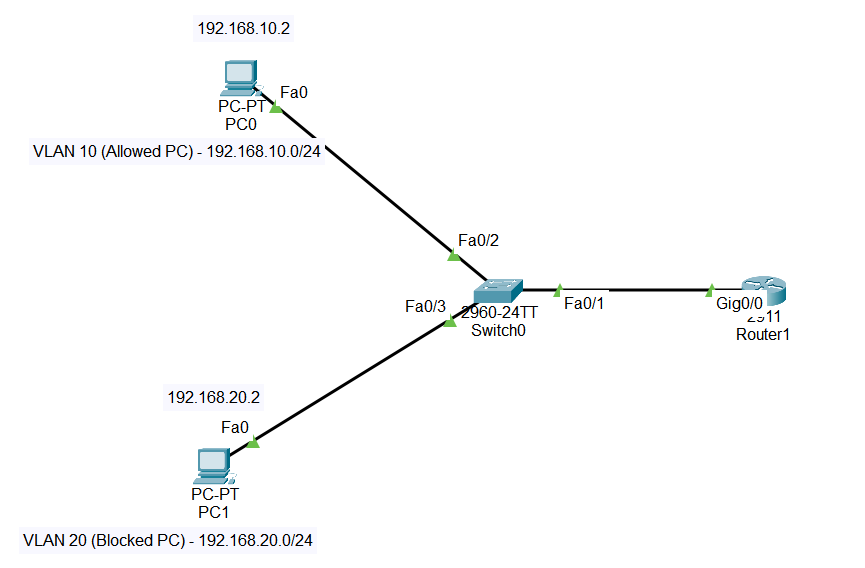
#### **5. Enabling Router Interfaces**

Ensure router interfaces are active using the **no shutdown** command on **GigabitEthernet0/1.0** and **GigabitEthernet0/1.20**.



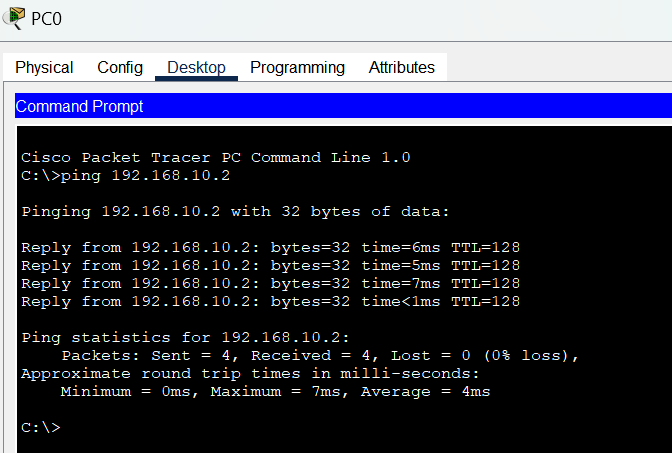
#### **6. Network Configuration in Cisco Packet Tracer**

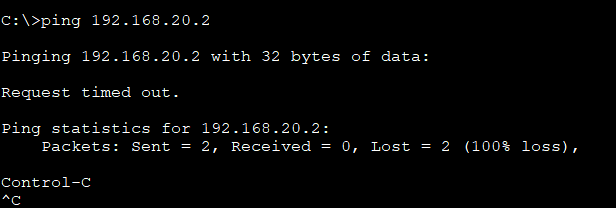
Simulate and test the ACL setup in **Cisco Packet Tracer** to validate the configuration.



#### **7. Testing Connectivity**

* **Ping test from allowed PCs:** Ensure PCs within the permitted IP range can successfully communicate.
* **Ping to restricted IP (192.168.20.2) fails:** Confirm that ACL rules are correctly blocking unauthorized traffic.





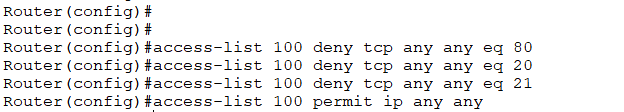
**Q12. Create an extended ACL to block specific applications, such as HTTP or FTP traffic.**

**Test the ACL rules by attempting to access blocked services.**

To restrict access to specific applications such as HTTP and FTP, follow these steps:

#### **1. Creating an Extended ACL**

Define an extended ACL to block **HTTP**, **FTP data**, and **FTP control traffic**.



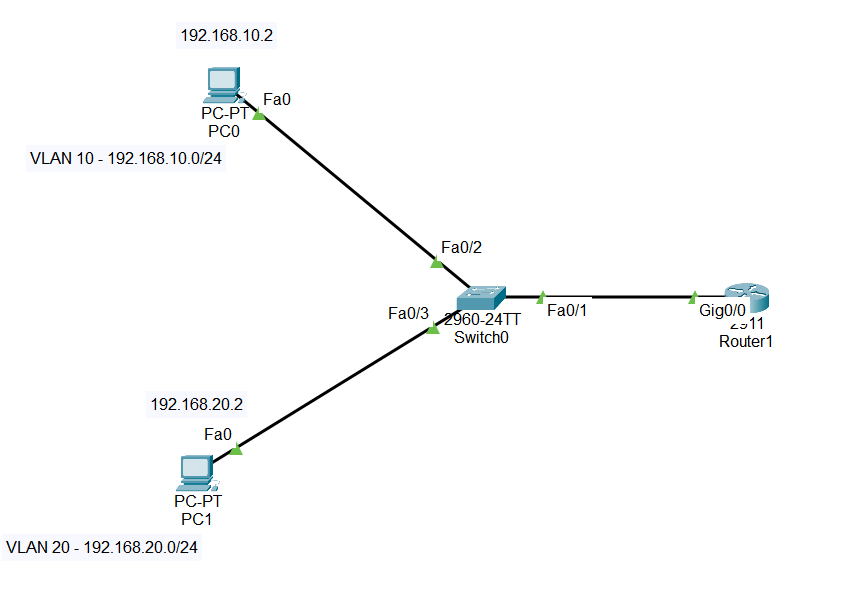
#### **2. Applying the ACL to the Router’s Interface**

Attach the extended ACL to the appropriate router interface to enforce traffic restrictions.



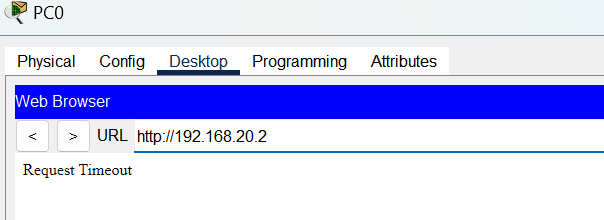
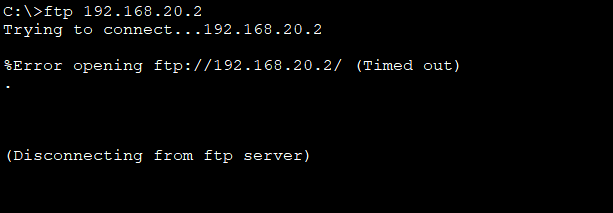
#### **3. Configuring the Network in Cisco Packet Tracer**

Simulate and test the ACL rules in **Cisco Packet Tracer** to verify functionality.



#### **4. Testing Blocked Services**

* **Attempting HTTP access from PC0 fails**, confirming that web traffic is restricted.
* **FTP access from PC0 fails**, ensuring that both FTP data and control connections are blocked.

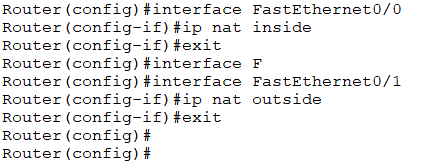


**Q13. Try Static NAT, Dynamic NAT and PAT to translate IPs**

To configure and test different NAT techniques, follow these steps:

#### **1. Assigning NAT Roles**

Define the NAT types to be used: **Static NAT, Dynamic NAT, and PAT (Port Address Translation)**.



#### **2. Configuring Static NAT**

Map a private IP to a specific public IP for one-to-one translation.



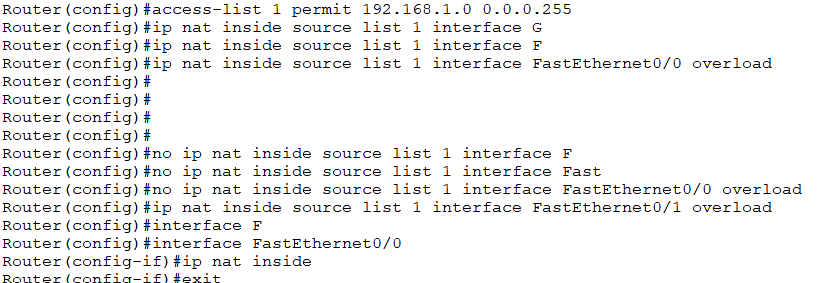
#### **3. Configuring Dynamic NAT**

Translate private IPs to a pool of public IPs dynamically.



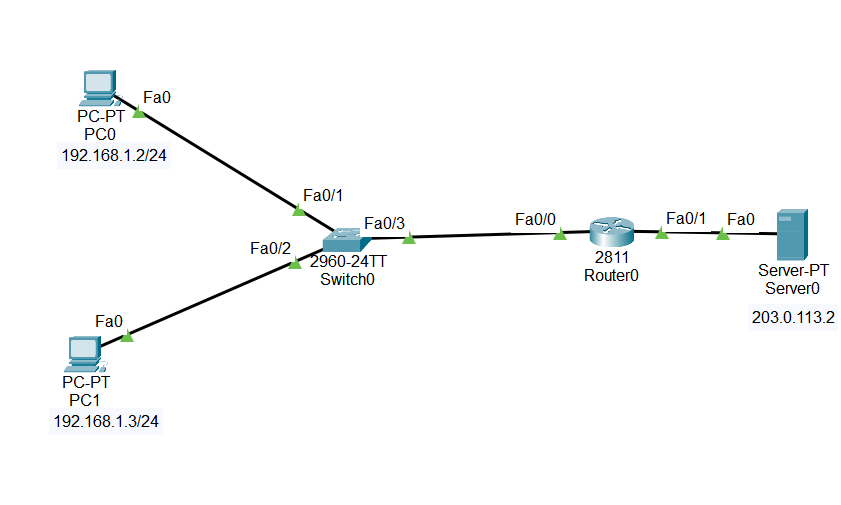
#### **4. Configuring PAT (Port Address Translation)**

Enable multiple private IPs to share a single public IP using different port numbers.



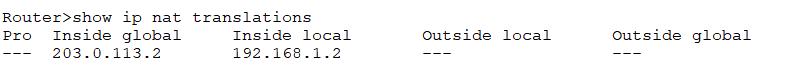
#### **5. Network Configuration in Cisco Packet Tracer**

Simulate and test the NAT setup in **Cisco Packet Tracer** to validate configurations.



#### **6. Testing NAT**

Verify that devices can access external networks using NAT translations.



**Q14. Download iperf in laptop/phone and make sure they are in same network. Try different iperf commands with tcp, udp, birectional, reverse, multicast, parallel options and analyze the bandwidth and rate of transmission, delay, jitter etc.**

### **Testing Network Performance Using iPerf with Various Options (Q14)**

To analyze network bandwidth, transmission rate, delay, and jitter, follow these steps:

#### **1. Setting Up iPerf on Devices**

Ensure that **iPerf** is installed on both a laptop and a phone, and they are on the same network.

#### **2. Starting the iPerf Server**

Run the **iPerf server** on a Linux machine:

iperf -s

#### **3. Running the iPerf Client on Windows**

Initiate a test from Windows using:

iperf3 -c <server\_IP>

#### **4. Testing Parallel Streams**

Execute parallel stream tests to analyze multi-threaded data transmission:

* **Parallel Streams (Test 1, 2, and 3)**
* **Parallel Streams Results in Linux**

#### **5. Bidirectional Testing**

Run bidirectional tests to measure **upload and download performance simultaneously**.

#### **6. Reverse Direction Testing**

* **Reverse direction test from Linux to Windows**
* **Reverse direction test from Windows to Linux**

#### **7. UDP Performance Analysis**

* **UDP test results in Linux**
* **UDP test for jitter and packet loss (Windows to Linux)**