

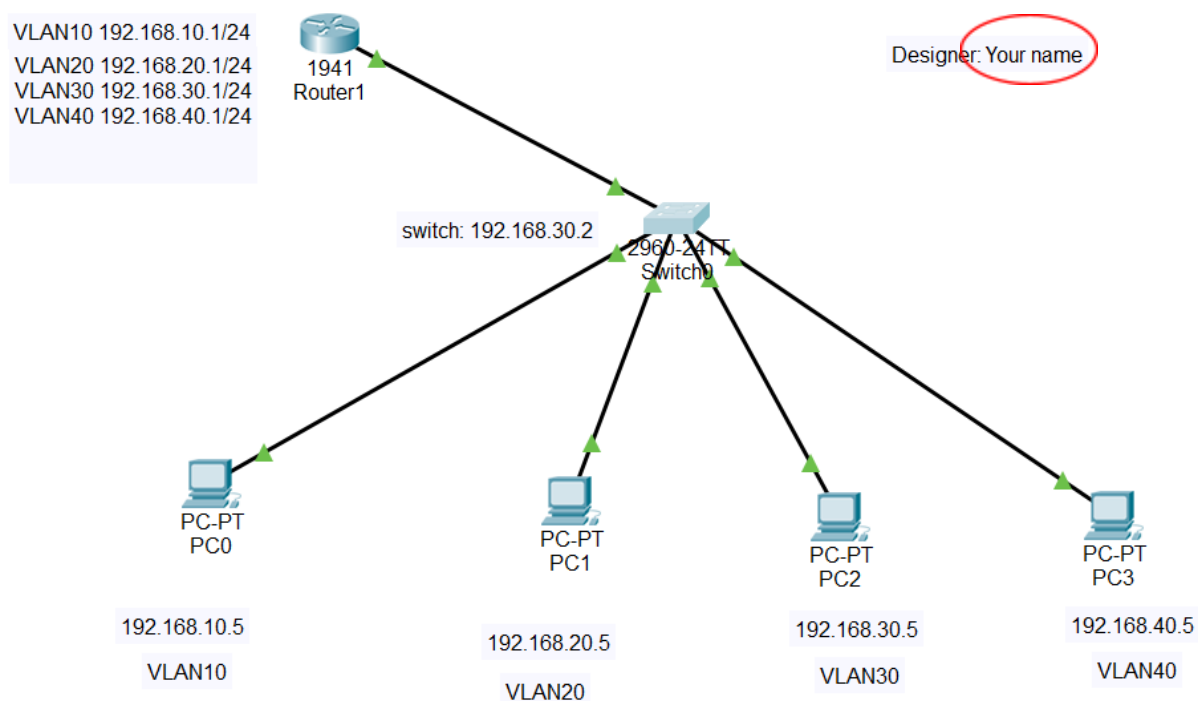
CSC/CIS 419 Network and Internet Applications

Networking Project: VLAN Design

Objective: Apply the knowledge of computer network to design VLAN networks

Create VLANs with the topology and the configuration as shown on the below figure. On the router, the interface g0/1 acts as the default gateway for VLAN10, VLAN20, VLAN30, and VLAN40. The switch has the ip address of 192.168.30.2 that belongs to VLAN30. PC0, PC1, PC2, PC3, and Switch0 are able to access each other.

Please refer to the video on BlackBoard for using cisco packet tracer.



Submission requirement (two files: pdf and word files are both fine while .zip is NOT accepted):

- (1) A well-organized user manual is expected to be submitted, including all configurations/command lines on various devices. The user manual intends to provide a quick start guide for network technicians so that the VLAN networks can be replicated by those who read and follow your instructions/command lines. The manual is expected to be clear and detailed.

Step 1: Select Router 1941

- Navigate to Network Devices and select Router 1941.
- Input the device.

Step 2: Select Switch 2960

- Go to Switches and choose Switch 2960.
- Input the device.

Step 3: Select PCs in End Devices

- Go to End Devices and choose PC.
- Input PCs 0, 1, 2, and 3.

Step 4: Connect PC0 to Switch0

- Go to Connections and select Copper Straight-Through.
- Connect PC0 (FastEthernet0) to Switch0 (FastEthernet0/1).

Step 5: Repeat for PC1-PC3

- Repeat Step 4 for connecting devices PC1-PC3 to Switch0 using connections FastEthernet 0/2, 0/3, and 0/4.

Step 6: Connect Switch0 to Router0

- Select Copper Straight-Through.
- Connect Switch0 (GigabitEthernet 0/1) to Router0 (GigabitEthernet 0/0).

Step 7-8: Configure IP Addresses on PCs

- Access PC0, set IP address to 192.168.10.5 and Default Gateway to 192.168.10.1.
- Access PC1, PC2, and PC3, set their IP addresses and Default Gateways accordingly.

Step 9-16: Configure Switch0

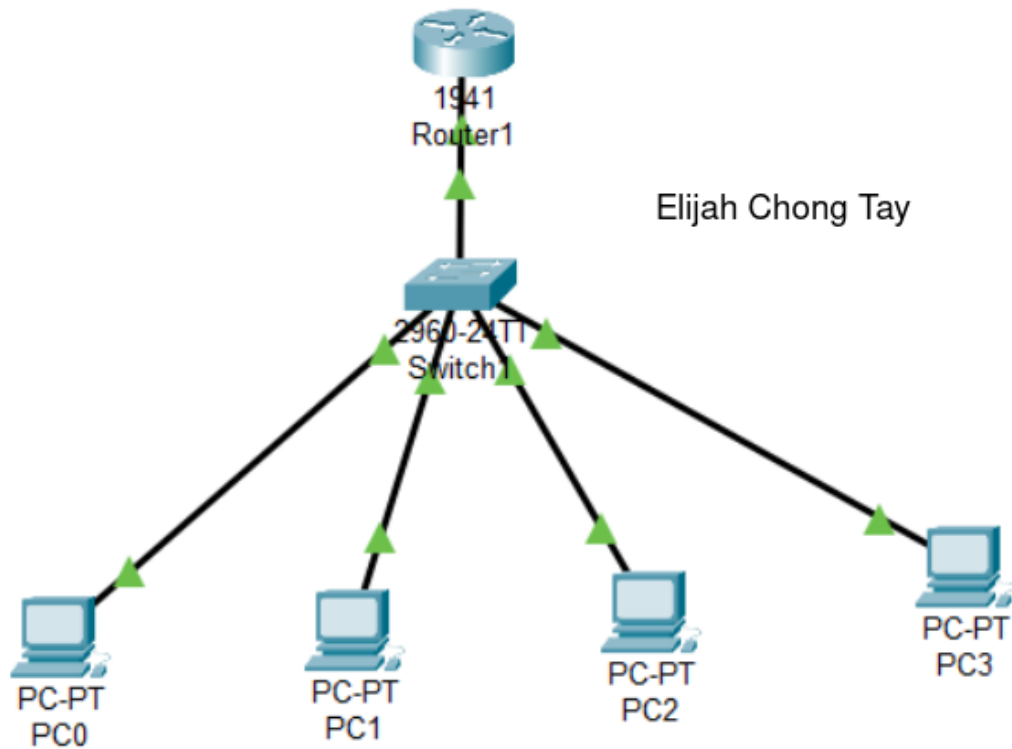
- Access Switch0 CLI.
- Execute commands for VLAN configuration, port settings, trunking, and save the configuration.

Step 17-22: Configure Router0

- Access Router0 CLI.
- Configure sub-interfaces for each VLAN on GigabitEthernet 0/1.
- Save the configuration.

Step 23: Completion

- Ping other devices from any PC to verify the setup.
- Save the configuration on Router0.



(2) A result file should contain:

- A screenshot shows the topology similar to the above diagram with your name.
- A screenshot shows that PC0 can successfully ping Switch0.
- A screenshot shows that PC1 can successfully ping Switch0.
- A screenshot shows that PC2 can successfully ping PC3.
- A screenshot shows that PC3 can successfully ping PC0.

```
C:\>ping 192.168.10.5

Pinging 192.168.10.5 with 32 bytes of data:

Reply from 192.168.10.5: bytes=32 time<1ms TTL=127
Reply from 192.168.10.5: bytes=32 time<1ms TTL=127
Reply from 192.168.10.5: bytes=32 time<1ms TTL=127
Reply from 192.168.10.5: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.10.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
C:\>ping 192.168.40.5

Pinging 192.168.40.5 with 32 bytes of data:

Reply from 192.168.40.5: bytes=32 time<1ms TTL=127
Reply from 192.168.40.5: bytes=32 time<1ms TTL=127
Reply from 192.168.40.5: bytes=32 time<1ms TTL=127
Reply from 192.168.40.5: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.40.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
Pinging 192.168.30.2 with 32 bytes of data:

Reply from 192.168.30.2: bytes=32 time<1ms TTL=254
Reply from 192.168.30.2: bytes=32 time<1ms TTL=254
Reply from 192.168.30.2: bytes=32 time<1ms TTL=254
Reply from 192.168.30.2: bytes=32 time<1ms TTL=254

Ping statistics for 192.168.30.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
Pinging 192.168.30.2 with 32 bytes of data:

Reply from 192.168.30.2: bytes=32 time<1ms TTL=254
Reply from 192.168.30.2: bytes=32 time=16ms TTL=254
Reply from 192.168.30.2: bytes=32 time<1ms TTL=254
Reply from 192.168.30.2: bytes=32 time<1ms TTL=254

Ping statistics for 192.168.30.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 16ms, Average = 4ms
```

Socket Programming Project

This project involves a simple server-client interaction via the terminal and localhost, demonstrating Java-based client-server communication using HTTP. The server listens on port 60000 and responds to client messages, updating a shared message on the server. The client sends a message to the server, and the server responds with a combined message.

Project Files

Server.java: Contains the server-side implementation.

Client.java: Contains the client-side implementation.

Instructions

Running the Server

Open the terminal or command prompt.

Navigate to the directory containing Server.java.

Compile the server code: `javac Server.java`.

Run the server: `java Server`.

The server will start running on `http://localhost:60000/hello`. You can access this URL in your browser.

Running the Client

Open another terminal or command prompt.

Navigate to the directory containing Client.java.

Compile the client code: `javac Client.java`.

Run the client: `java Client`.

The client will prompt you to enter a message. Enter a message, and the client will send it to the server. The server will respond with a combined message.

Project Notes

The server processes POST requests at the `/hello` endpoint.

The client sends a user-defined message to the server.

The server combines and displays the client's messages.

The server can be gracefully stopped using a shutdown hook.

Example

Start the server.

Open the client and enter a message.

The server will display the client's message along with the server's response.

Dependencies

Java SDK 8 or higher.