

Small Town Networking Project

This is a simulation of a small town's network designed to simulate realistic protocols and setups of a network on Cisco Packet Tracer.

Planning

The planning for this was relatively simple. To supplement my learning of internet protocols, I wanted to have a plethora of different networks that could act as a simulation of how a town could look while being able to experiment with different network setups and configurations.

The initial setup was as follows:

- A law firm with three LANs with wired ethernet configurations.
- A hospital with four LANs, three with cabled ethernet configurations and one wireless guest network.
- A school with four LANs, three with wireless configurations and one with cabled ethernet configuration.
- A tech startup with four LANs, two cabled ethernet configurations, one wireless configuration, and one reserved for a server that ran protocols to simulate internet access.
- A wireless home network with one LAN.

Each router would then be connected using a ring topology to allow access to multiple LANs.

Routing and Subnetting

Most networks were configured using the "router-on-a-stick" method because each ISR4331 router didn't have enough ports to support multiple connections. The router would be connected to a 2960 IOS15 switch through one router port, and the switch would be connected to multiple devices to account for the lack of router ports.

Ethernet Networks

The steps to form an Ethernet connection with static IP address assignment are as follows:

1. Connect the router to the switch through a GigabitEthernet or FastEthernet port using a straight-through copper cable.
2. Connect the switch to the device through a FastEthernet port using another straight-through copper cable.

3. Open the port connecting the router to the switch on the router's CLI and assign it an IP address:

```
Router> enable
Router# configure terminal
Router(config)# interface GigabitEthernet0/0/0
Router(config-if)# no shutdown
Router(config-if)# ip address 192.168.2.1 255.255.255.0
```

4. Assign the PC's interface an IP address from its configuration tab (for example: IPv4 address 192.168.2.2, subnet mask 255.255.255.0).
5. Ping the router from the PC's command prompt to verify connectivity:

```
ping 192.168.2.1
```

If the PC receives ping replies from the router, the Ethernet connection is successfully established and working as expected.

Wireless Networks

Wireless networks were assigned a separate VLAN and broadcasted the network through a WAP (the AccessPointPT device on Cisco Packet Tracer).

The steps to form a wireless connection with static IP address assignment are as follows:

1. Connect the router to the switch through a GigabitEthernet or FastEthernet port using a straight-through copper cable.
2. Connect the switch to the WAP through an Ethernet port using another straight-through copper cable.
3. Open the port connecting the router to the switch on the router's CLI and assign it an IP address:

```
Router> enable
Router# configure terminal
Router(config)# interface GigabitEthernet0/0/0
Router(config-if)# no shutdown
Router(config-if)# ip address 192.168.5.1 255.255.255.0
```

Enable DHCP on the router for the wireless network:

```
Router> enable
Router# configure terminal
Router(config)# ip dhcp pool LAW_GUEST
Router(dhcp-config)# network 192.168.5.0 255.255.255.0
Router(dhcp-config)# default-router 192.168.5.1
Router(dhcp-config)# dns-server 8.8.8.8
```

(Note: The 8.8.8.8 dns-server here acts as a placeholder until an actual DNS server is configured later in the project)

4. Ping the router from the PC's command prompt to verify connectivity:

```
ping 192.168.5.1
```

If the PC receives ping replies from the router, the wireless connection is successfully established and working as expected.

VLAN Segmentation and IP Assignment

VLANs (Virtual Local Area Networks) allow network administrators to logically separate traffic on a single physical switch. This improves security, reduces congestion, and makes network management easier. Each VLAN acts as an independent subnet, preventing broadcast traffic from crossing between networks such as administrative LANs and guest Wi-Fi.

The following steps describe how to create VLANs, assign ports, and configure router subinterfaces for inter-VLAN communication.

Create VLANs on the switch

```
Switch> enable
Switch# configure terminal
Switch(config)# vlan 2
Switch(config)# name HR
Switch(config)# exit
```

Assign switch ports to VLANs

```
Switch(config)# interface FastEthernet0/1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 2

Switch(config)# interface FastEthernet0/4-7
```

```
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 3
```

Configure router subinterfaces for inter-VLAN routing (Router-on-a-Stick)

```
Router> enable
Router# configure terminal
Router(config)# interface GigabitEthernet0/0/0.2
Router(config-subif)# encapsulation dot1Q 2
Router(config-subif)# ip address 192.168.2.1 255.255.255.0
Router(config-subif)# exit
```

(Note: It's generally best practice to number the subinterfaces after the VLAN number for clarity and convention)

Notice how the VLAN configuration with subinterfaces follows nearly the same steps as configuring a whole interface. Splitting the switch into VLANs then configuring each network is usually faster than configuring the network and setting up VLANs. This is also true for dhcp pools, and the subinterface that is assigned the dhcp pool's gateway router's IP address acts as the main router.

Test VLAN connectivity

From a device within each VLAN, ping the other VLAN.

Ex. from the 192.168.3.0/24 network,

```
ping 192.168.2.2
```

From the 192.168.2.0/24 network,

```
ping 192.168.3.2
```

If devices within each VLAN receive IP addresses and successful ping responses, the VLAN configuration and routing setup are functioning correctly.

Routing and Subnetting

Initially, the network was going to be designed as a ring topology, with each router being connected to the one before it and next to it. There would be a server hosting DNS and HTTP/HTTPS in order to simulate the Internet connected to one of these servers. However, this

isn't how the real Internet works, and if one router stops working, the entire network collapses. Instead, the improved network operates on a star topology, using a central router to forward information between routers instead of direct router-to-router connections. This central router is meant to be the "Internet" and simulate multiple hops across the Internet to reach the desired router.

Internal Configuration:

The configuration on the internal router which hosts the private 192.168.x.x networks is simple. Only the following command needs to be passed to the CLI:

```
Router>enable
Router# configure terminal
Router(config)# ip route 0.0.0.0 0.0.0.0 79.52.164.1
```

This command sends traffic to any unknown network (0.0.0.0/0) to the central router, 79.52.164.1. This configuration is necessary on every internal router.

Edge Configuration:

The edge router meant to simulate the Internet needs to know the networks underneath each of the internal routers and the WAN address of the router. This edge router functions as the ISP, managing external routing and simulating how local organizations connect to the Internet. Let's say, for example, that all the routers exist on the network 79.52.164.0/24. For this example, let's use the Law Firm Topology's router, which hosts the networks 192.168.2.0/24, 192.168.3.0/24, 192.168.4.0/24, and 192.168.5.0/24. The Law Router's WAN address is 79.52.164.8. The following commands need to be executed on the edge router's CLI.

```
Router>enable
Router# configure terminal
Router(config)# ip route 192.168.2.0 255.255.255.0 79.52.164.8
Router(config)# ip route 192.168.3.0 255.255.255.0 79.52.164.8
Router(config)# ip route 192.168.4.0 255.255.255.0 79.52.164.8
Router(config)# ip route 192.168.5.0 255.255.255.0 79.52.164.8
```

This configuration must be performed on the edge router's CLI for all the internal routers and their respective private networks. Because of the intense cabling, the edge router uses the "router on a stick" method to connect all the cables to a switch, which then connects everything back into the edge router. Once the cabling is established, the entire network should be able to communicate. Ping tests can be run from one private network under one internal router to

another private network under another. For example, the following executed on 192.168.2.2's command prompt under Law Router tests connectivity to 192.168.40.3 under Tech Router.

```
ping 192.168.40.3
```

Running tests like this from each subnet to every other subnet confirms the established communication between the entire small town network.

Server Configuration

There are two essential services required for a device to connect to a website: **HTTP** and **DNS**.

- **HTTP (Hypertext Transfer Protocol)** handles the actual connection to the web server and loads the contents of the requested website.
- **DNS (Domain Name System)** maps human-readable domain names (like <http://google.com>) to their corresponding IP addresses, eliminating the need to memorize numerical addresses.

Configure the HTTP

For this simulation, I used the Server-PT device on a packet tracer. The first step is enabling the HTTP service if not enabled by default. Packet Tracer comes with a pre-loaded HTML template, but in order to change the contents of the website, all one needs to do is edit the contents of `index.html`.

Configure the DNS

After enabling the DNS service on the server, it's time to map a domain name to an IP address. In order to improve the searchability of the website, I mapped both <http://readme.com> and readme.com to the 79.52.164.250 server. Note that this server is on the same network as the router system. After configuring the DNS on the server, each device needs to point to 79.52.164.250 as its DNS server before being able to access the website through the domain name. In order to do this, I went back and set each device's DNS server to 79.52.164.250 using the config tab on the PCs and Laptops. For DHCP servers, entering DHCP config on the router's CLI and setting the dns-server to 79.52.164.250 is necessary.

```
Router(config)# ip dhcp pool LAN_POOL
Router(dhcp-config)# dns-server 79.52.164.250
```

Once the DNS server is properly configured, every device on every network should be able to access the domain name (readme.com in this case). This can be more efficiently tested by going into a web browser on a device in every private network and attempting to connect to readme.com.

Conclusion

The purpose of this project was to design and simulate a small town's interconnected network infrastructure using Cisco Packet Tracer. By building multiple organizations and connecting them through static routes, VLANs, and a simulated Internet, the network reflects how real-world communities manage communication and data transfer between institutions.

Each organization, such as the law firm, hospital, school, and tech startup, was configured with distinct Ethernet and wireless LANs. VLAN segmentation improved traffic control and security, while static routing between routers enabled full communication across all subnets through a central Internet router. After setting up HTTP and DNS services, users could access a locally hosted web page using a custom domain name, completing the simulation of realistic web functionality.

This project reinforced my understanding of networking concepts such as IP addressing, DHCP configuration, VLAN management, inter-VLAN routing, and DNS mapping. It also highlighted the importance of efficient topology design to prevent single points of failure and maintain reliable communication.

In the future, the network could be expanded with NAT, ACLs, or dynamic routing protocols like OSPF to enhance scalability and fault tolerance. Overall, this project successfully brings together multiple networking elements to create a cohesive and functional simulation of a small town network.