Week1

Network Theory

Networking Components

Hardware- can take data onto and take data off the medium
A medium to carry the data between the devices
Protocols (rules) of how the two devices will communicate

Nodes/Hosts: Anything connected to a network
Redistribution Points: Transfer data on the path
Endpoints: Source or destination

Node Functions

Servers

Any node that shares resources and responds to requests can be called a server. All computers generally function as servers in some way. However, when we use the word “server,” we’re typically talking about a computer that has been designed to provide services to other devices. They’re usually kept in locked rooms away from the users.
Servers supply central resources. These resources can include applications, files or printers and other hardware. A server can be dedicated to one specific function, or it can serve general needs. And multiple servers of more than one type can exist on the same network.
Because other devices depend on the services of the server, servers usually have redundant (duplicate) hardware components. That way, even if something breaks, the server can continue to run. They also usually have special operating systems. The most common server operating systems in use today are Microsoft Windows Server ® and Linux.

Clients

A client is a network computer that uses the resources of servers. The client computer can also perform its own tasks and processing. All computers generally function as clients at some point. However, when we use the word “client,” we’re typically talking about a computer that has been designed to be used by end users. Clients are often called desktops or workstations. They usually run operating systems that are more responsive to users. Client also implies the computer is used in a business. The most popular client operating systems are Microsoft Windows ® and certain distributions of Linux.
Suppose you have a printer attached by a cable to your computer. If you allow someone else in your home to print to that printer, technically you’re the server. The other computer is the client. But usually,

these words describe business environments where the two devices are specially configured for what they do most of the time.

Peer Computers

A peer is a computer that acts as both a server and a client to other computers on a network. Peer computing is most often used in smaller networks that don't have a dedicated server. Although, peers can belong to networks with servers.
Peer computers run client operating systems. The key difference between clients and peers is whether they have a security relationship with the server. If users that have an account on the server can log in on the workstation, it's a client. If the user needs to have an account on the workstation, then it's a peer. In the above scenario, where you shared your printer with a family member, your computer is functioning as a peer.

Host Computers

A host computer is a central computer system that performs storage and processing for other devices. On a host-based network, the host computer does all computing. It then returns the data to the end user's terminal. Host computers are often referred to as mainframes.
In the early days of networking, all computers were hosts. The hosts were then joined together in the early research networks that became the Internet. As the TCP/IP protocol became popular, and personal computers joined the networks, the term host became generalized. Now "host" is used to describe to any node on a TCP/IP network.

Terminals

A terminal is a specialized device on a host-based network. Users enter data into the terminal. The terminal sends the data to a host for processing. The host sends the results back to the terminal. Terminals are often called "dumb terminals." Unlike clients, they have no processor or memory of their own. They're usually just a keyboard and a monitor. Standard client computers that need to interact with host computers can run software called a terminal emulator so that they appear as terminals to the host.

Network Categories

LANs

When it comes to types of networks, the terms can be confusing. The nature of networking has changed quite a bit since these terms were invented.
A Local Area Network (LAN) implies a self-contained network. LANs exist in small areas, such as a single building, floor, or room. In a LAN, all nodes are directly connected with cables or short-range

wireless. LANs do not need any outside technology, like an Internet Service Provider (ISP), to function. Due to their smaller size, LANs have faster speeds than other network types. Most modern LANs use a technology called Ethernet. You will learn more about Ethernet later in the course.
Instead of “LAN,” professionals might refer to a LAN as the “local network.”
If you’re talking about a computer, “local” means “contained in the computer itself.” If you’re talking about a network, “local” means “connected to the same network.” This might refer to the whole LAN. Or it could mean “all the nodes that can talk to each other without needing a router.” Routers are devices that connect two or more different networks and can pass information between them.
Typically, LANs are supported by LAN Administrators. They manage and update the local network. The administrator’s job includes servicing hardware, cabling and software. They may perform installations and deployments, upgrades, and troubleshooting. To be a LAN administrator, you need a broad range of skills and knowledge about networking, software and hardware.

WANs

A Wide Area Network (WAN) is a network that spans a large area. WANs often cross countries or continents. Typically, WANs connect multiple LANs and other networks. They use long-range transmission media provided by telecommunications companies. WANs can be private, which means that they belong to one company. Or they can be public, meaning they can be used by anyone. The Internet is a public WAN.
When multiple networks form a larger network, we often call them subnetworks, subnets or segments. In that case, the “local network” is the one you’re using. The other networks are called “remote.” When messages travel through multiple networks, the connections are usually made by routers. That’s why we say that messages (traffic) are “routed” through a network.
Typically, WANs are maintained by WAN Administrators. They usually address more complex technical issues than LAN administrators. They tend to focus on resolving network issues rather than user issues. A WAN administrator typically performs the following duties:

- Designs and maintains the connections between remote segments.
- Develops and troubleshoots routing structures.
- Works with both voice and data systems.
- Develops scripts to automate administrative tasks.
- Works on security issues and helps implement recovery schemes.
- Plans, tests, and implements hardware and software upgrades.

More N/w Terminology

The Internet, publicly owned and operated, is the largest WAN. It links virtually every country in the world. Here is a brief history of the Internet.

1957 **The United States government forms the Advanced Research Projects Agency (ARPA). The goal is to make the US a leader in military science and technology.**

1962 The US Air Force conducts a study on how to keep control of missiles and bombers after a nuclear attack. The recommended solution is a decentralized military research network.

1969 ARPA launches ARPANET. ARPANET Initially connects only four nodes owned by universities. By 1981, there are 213 computers with another node joining every 20 days.

1973 To carry data across long distances, ARPA wants to connect ARPANET to radio and satellite. This will not be possible without a common protocol. Vinton Cerf and Bob Kahn invent TCP/IP. Each node can be contacted at a unique address, known as an IP address.

1977 The first data is sent from the United States across wired, radio and satellite networks to the UK. The Internet is born!

1983 ARPANET invents the Domain Name System (DNS). DNS matches domain names to IP addresses. This allows users to easily contact specific computers on the Internet without memorizing IP addresses.

1989 Tim Berners-Lee invents the World Wide Web to solve the problem of how to find specific documents on the Internet. Users access web pages using web browsers by entering a Uniform Resource Locator (URL). Once the user opens a web page, links on the page allow the user to find the next web page.

Standard N/w Models

Overview of Network Models

A network model describes how the nodes on a network are interact. Network models vary based on how communications and processing are centralized or distributed.
The three network models we will be discussing are:

- Centralized
- Client/Server
- Peer-to-peer

These network models focus on the way the different nodes accomplish the primary objectives of the network. But they're not the only way we describe a network.
Networks have a physical topology. This describes how the nodes are physically connected. They also have a logical topology. This describes how the data flows through the network. For example, Ethernet (the most common technology used for LANs) is usually wired together in a star topology. Each device has a wire connection to a central point, usually a switch. The data in a wired Ethernet network uses a bus topology. In a bus network, all the nodes see all the traffic. Thus, we can describe Ethernet as a "physical star, logical bus."

But many professionals work their whole careers and don't have to worry about either the physical or logical topology of their networks. The roles of the nodes on the network are always important. When you enter a new network, you will almost always want to know how processing is being handled. If there's a problem, knowing the network model helps identify where to look for the solution.

Centralized

A centralized network is a computer network that uses a host. The host controls all network communication. It manages all the processing and storage. Users connect to the host using terminals or terminal emulators. If someone refers to a mainframe, they are probably implying a host.

Centralized networks deliver high performance. They allow centralized network management. This makes the network easier to support and more secure. The problem is that they're usually expensive.

The first computer networks were centralized. At that time, all computers were large and expensive. Using terminals to connect to the host allowed more than one user to use the one big computer. But this model is not limited to "legacy" (old, obsolete) environments or mainframes.

Cloud-based computing could also be described as centralized. Clouds usually have huge amounts of storage and processing power. They can be accessed by millions, maybe even billions, of clients. Clients typically access the cloud via a browser. If you have a public email address through Gmail, Outlook, or Yahoo, you're using cloud-based computing.

There is also a new form of centralized computing called Virtual Desktop Infrastructure (VDI). In VDI, employees don't have workstations at their desks. Instead, each employee has a terminal that allows them to access a virtual desktop. The desktop operating system runs on a computer somewhere else in the company. This can save the company money because they only need one powerful server to host the virtual desktops. If the terminal at someone's desk fails, the employee can move to any other terminal. There is only one problem with this model. If the employee only has a terminal, and the host (or the network) fails, the employee cannot work at all.

Centralized computing always simplifies management and security. The problem with all centralized computing is that it makes a failure of the network or central computer harder to survive. Whenever the failure of one device can disrupt a whole system, we call that device a Single Point of Failure (SPoF).

Client/Server

A client/server network is a network in which servers provide resources to clients. Both the clients and servers have their own local processors and storage. Using servers allows centralized management and security. Clients perform basic end-user tasks on their own. Because some of the processing happens on the client, the servers don't need to be as expensive as hosts. It also allows administrators

to place the processing power closer to where it's needed. Tasks that don't need a lot of processing power can be done on the clients. Tasks that require more resources can be done on the servers.

In a client/server network there's usually at least one server in charge of central authentication. That server hosts a database of usernames and passwords. The users can log in to any client in the network. The client transmits the information to the server. Authentication happens when the server verifies the identity of the user. The user proves their identity by sending a valid combination of a username and password or some other information to prove their identity.
Typically, servers aren't as powerful or expensive as host computers. That means companies can buy multiple servers for the same amount of money (or less) than needed to buy one host. Having multiple servers allows the company to achieve fault tolerance. Fault tolerance literally means a system that can tolerate a "fault" (failure). For practical purposes, fault tolerance means there is a backup that can takeover when something fails with little to no interruption.
The Internet is largely built on client/server concepts.

Peer-to-Peer

A peer-to-peer network does not have centralized control. Resource sharing, processing, and communications happen at all computers. All clients on the network are equal in terms of supplying and using resources. Each workstation authenticates its users.

Peer-to-peer networks are easy and inexpensive. However, they are only practical in small companies. A peer-to-peer network is more commonly referred to as a workgroup. More recently, the industry uses the term SOHO (Small Office Home Office).

In peer-to-peer networks, users need a username and password on each computer. Suppose you created a peer-to-peer network in your home. You're logged in on your laptop. You would like to print to a printer connected to a desktop in a study. If the study computer has a user with the same username and password as the laptop, you will be able to print with no issues. If the username or password is different, you would need to login to the study computer to print.

Effectively, each user needs the same username and password at each machine. This makes running a peer-to-peer network difficult. As the network grows, it gets more difficult.

There are other types of networks on the Internet that are called peer-to-peer. Torrent files and bitcoins are two examples. Both could also be described as "distributed computing."

The term "distributed computing" implies many inexpensive devices that belong to lots of different people. Distributed computing uses multiple "small" devices for tasks that would normally require a large, expensive computer. It can even be used for tasks beyond the resources of any one computer!

Torrent files can speed up file sharing. Torrents divide each file into small pieces. They also record which computers in the network have a copy. If multiple users have a copy of the file, someone who wants to download it can get pieces from more than one user at the same time. This allows the downloader to download faster because each host only processes part of the request. Unfortunately, torrents are often used to illegally share media.

Bitcoins were developed as an alternative to central banks. The idea is that anonymous participants can use the bitcoin system. Balances and transactions are recorded in a central, public ledger. Because there are so many copies of the ledger, it is theoretically impossible to create a false entry in it. Here are some good links if you're interested in bitcoins or torrents, but you are not required to read the articles.

Network Theory Lab

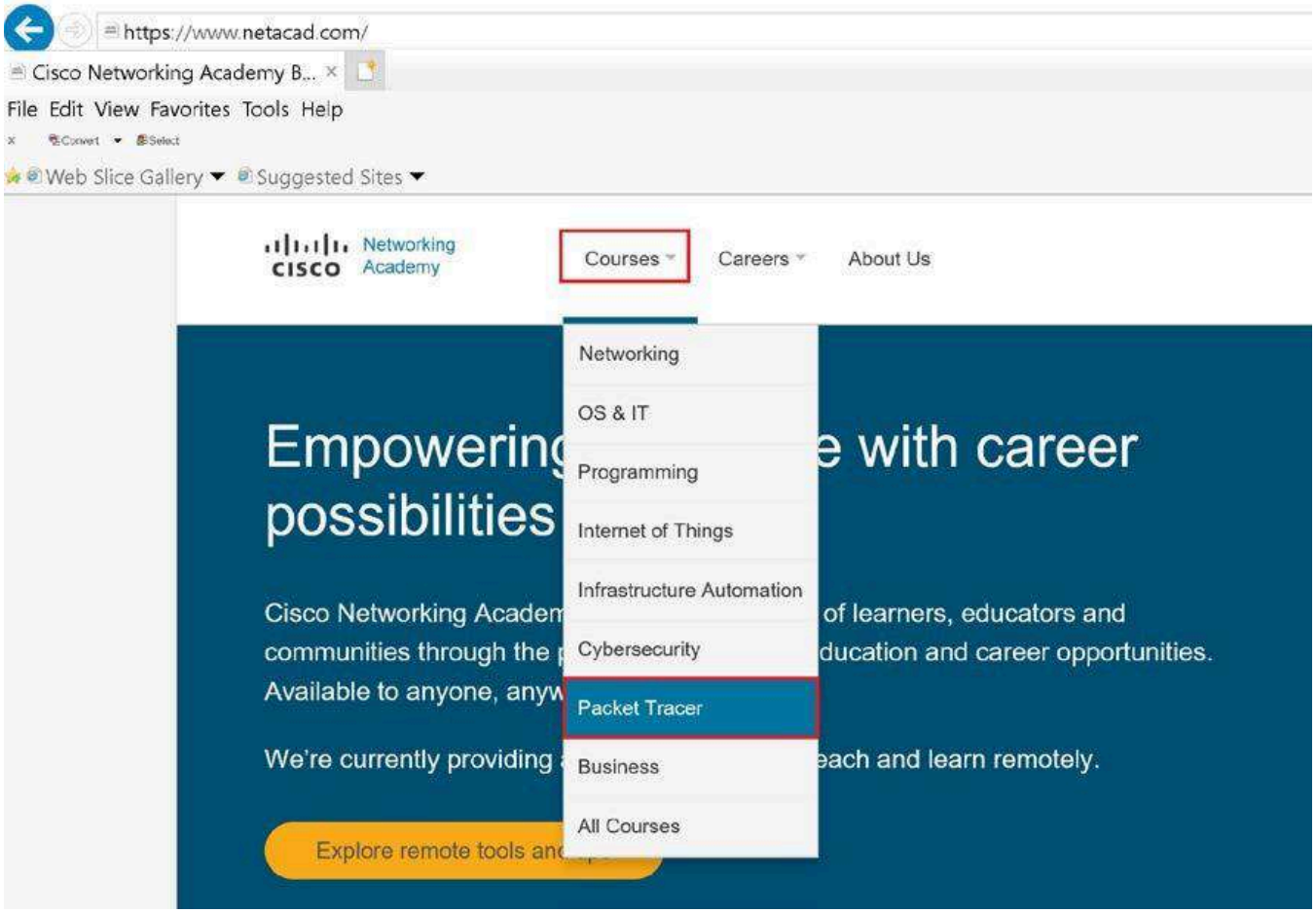
Install Packet Tracer

In this course we will use a free utility developed by Cisco for training students in networking. The utility is called Packet Tracer. You can use this to explore network infrastructure in a way that would be difficult without a lot of equipment. Before we can explore this utility, you will need to download and install it. **PLEASE NOTE: If the instructions for this lab are slightly off (web pages do change!) then please observe the instructions on the site for guidance.**

TASK A

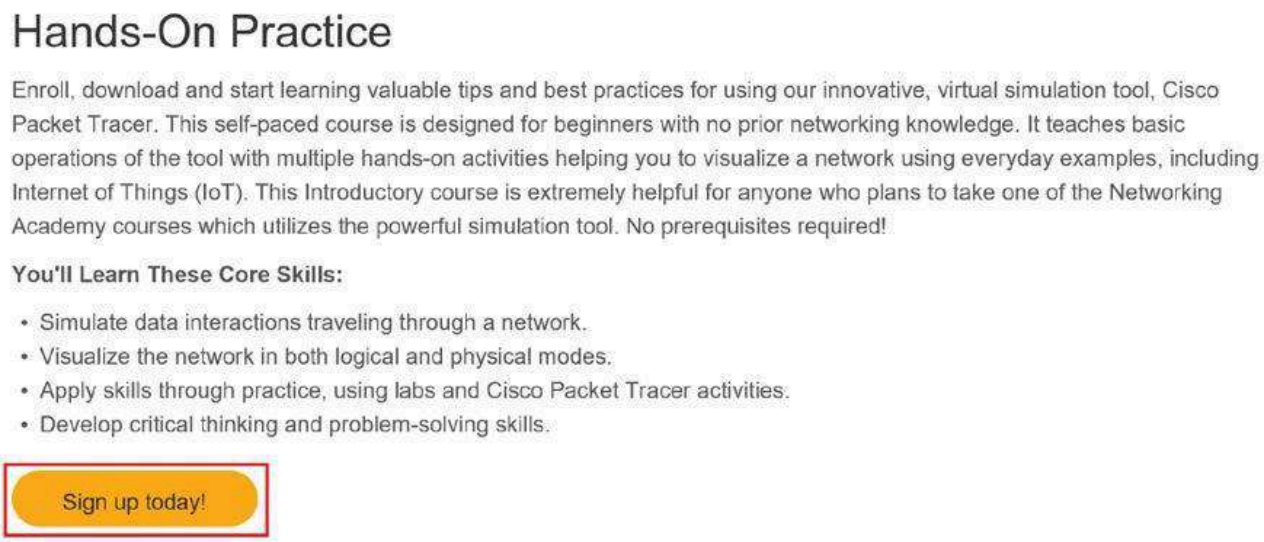
First, you will need to sign up for a course on the Cisco Networking Academy site:

1. Open a browser and navigate to **https://www.netacad.com/**
2. Select the **Courses** menu, and then click **Packet Tracer**.

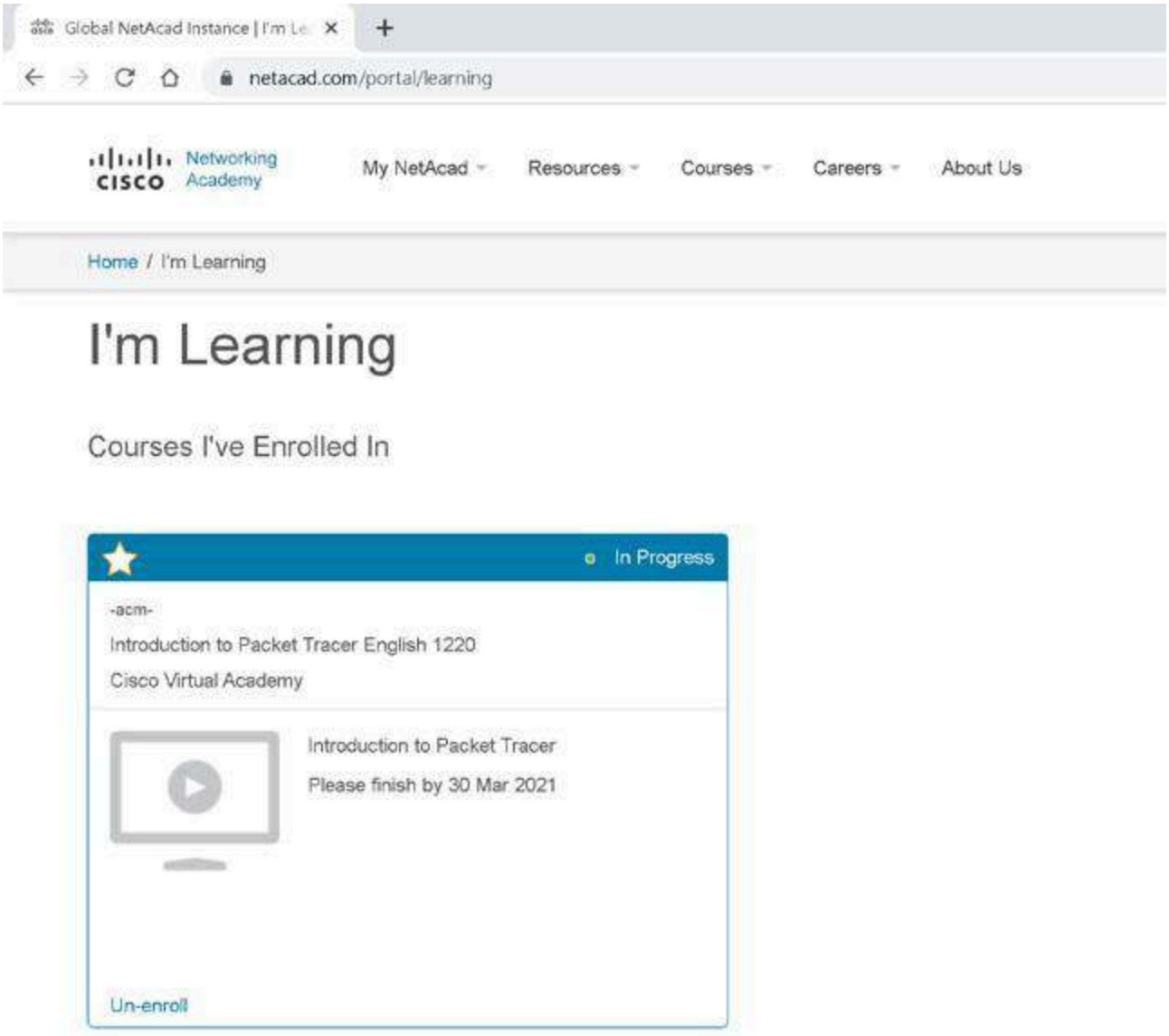


3. Click the **Introduction to Packet Tracer course** hyperlink.

4. Click the **Sign up today!** button.

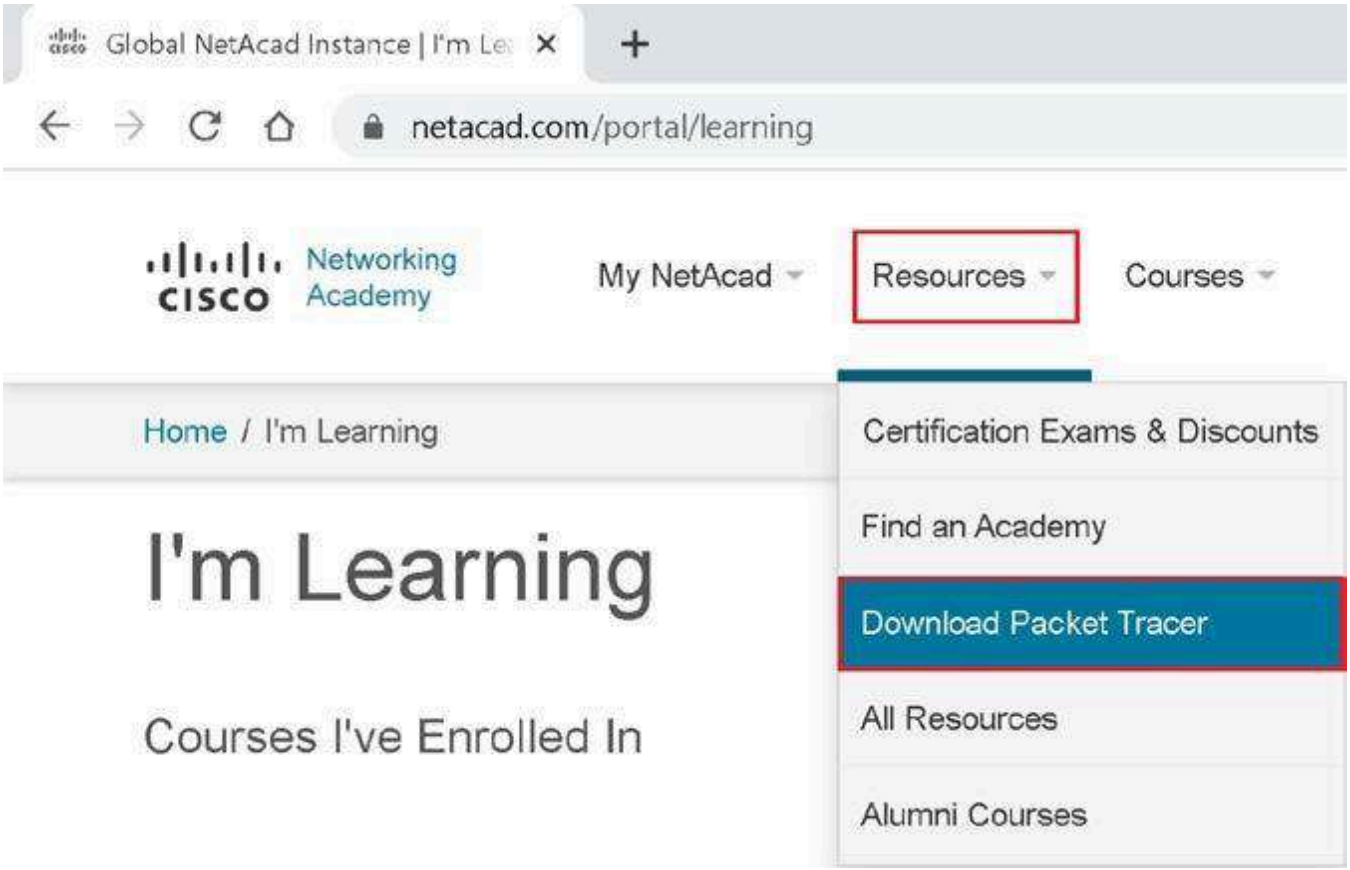


5. Follow the instructions to sign up for the academy.
6. Sign-in to the Academy.

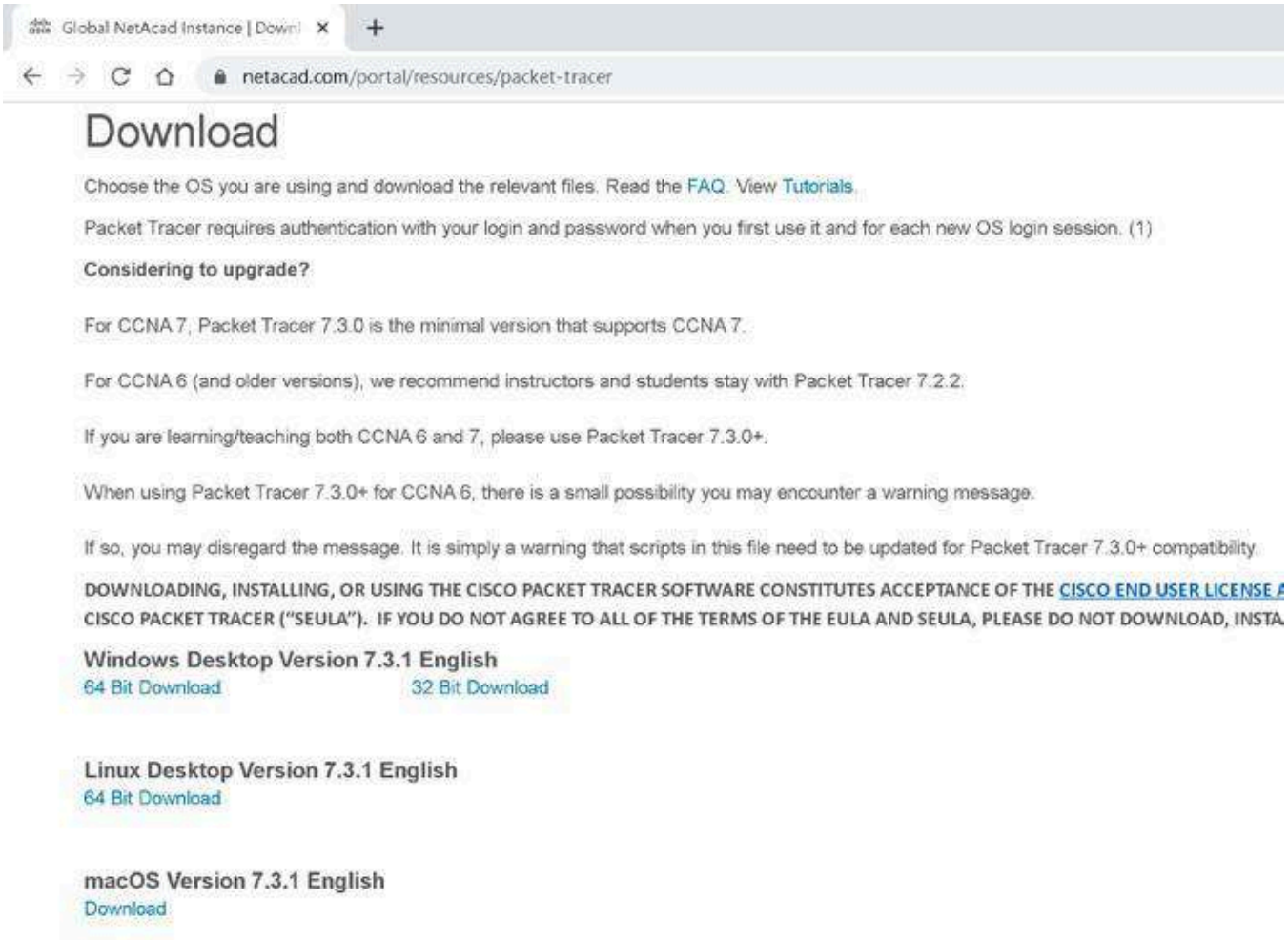


TASK B

Next, you will need to download the Packet Tracer software:
1. From the **Home** page of your academy account, open the **Resources** menu, and then click **Download Packet Tracer**.



2. On the **Download Cisco Packet Tracer** page, scroll down and select the appropriate version to download.



TASK C

Once you have downloaded the software, it needs to be installed. (NOTE: You are installing the software with all of the default options.)

1. Double-click the file you downloaded.
2. In the **License Agreement** dialog box, click the **I accept the license** radio button, and then click **Next**.
3. In the **Select Destination Location** dialog box, click **Next**.
4. In the **Select Start Menu Folder** dialog box, click **Next**.
5. In the **Select Additional Tasks** dialog box, click **Next**.
6. In the **Ready to Install** dialog box, click **Install**.
7. Click **Finish**.
8. If the application doesn't launch, start the **Packet Tracer** application.
9. In the **Would you like to run multiuser when application starts** dialog box, click **Yes**. If you get a firewall alert, click **Allow access**.
10. When prompted, log into your academy account. The **Packet Tracer** window should now appear.

Create a Simple Network

At a bare minimum, networking requires two devices that have:

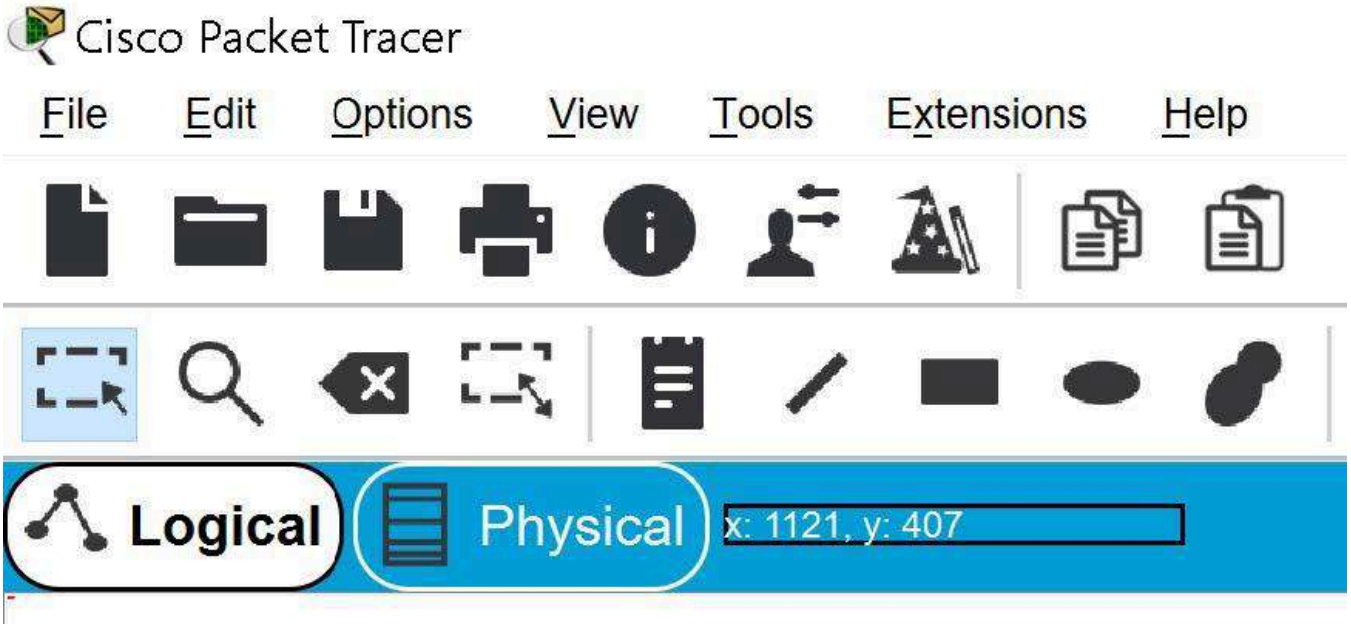
1. Hardware that can put data onto and take data off the medium.
2. A medium to carry data between the devices.
3. Protocols (rules) for how the two devices will communicate.

In this exercise, you will use Packet Tracer to create a simple network.

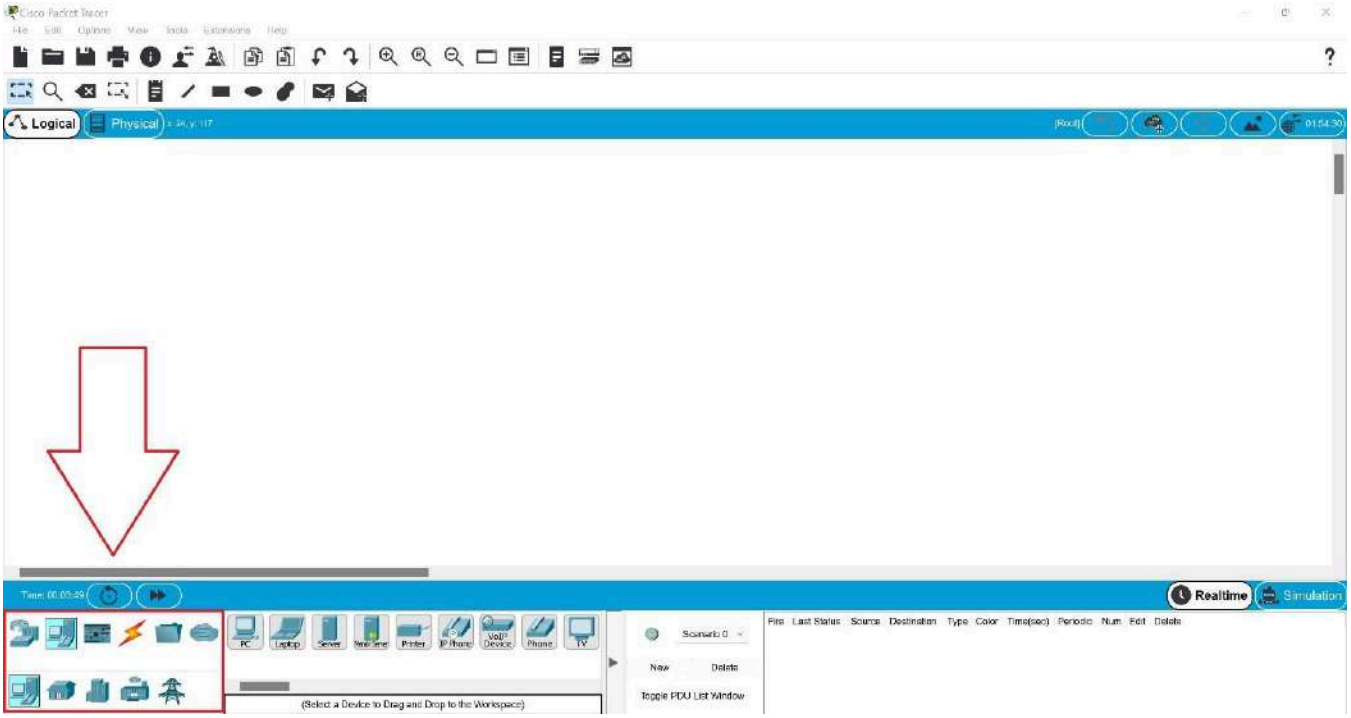
TASK A

Let's start by adding two devices to our network:

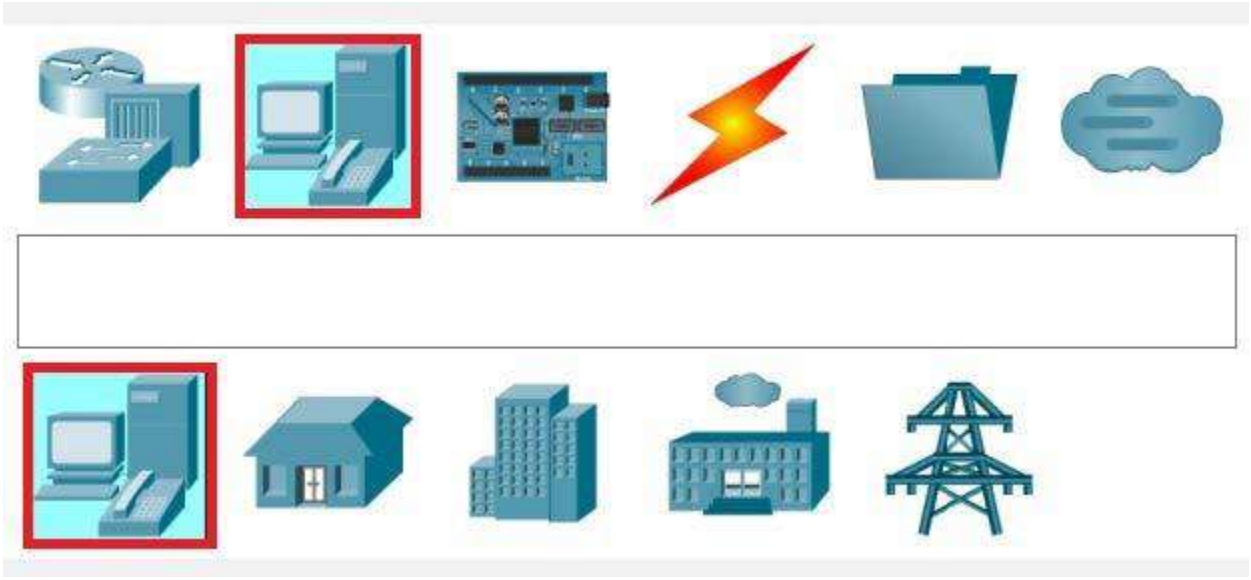
1. If the **Packet Tracer** application is not open, launch **Packet Tracer** and log in if necessary.
2. In the upper left corner, make sure to select the **Logical** button.



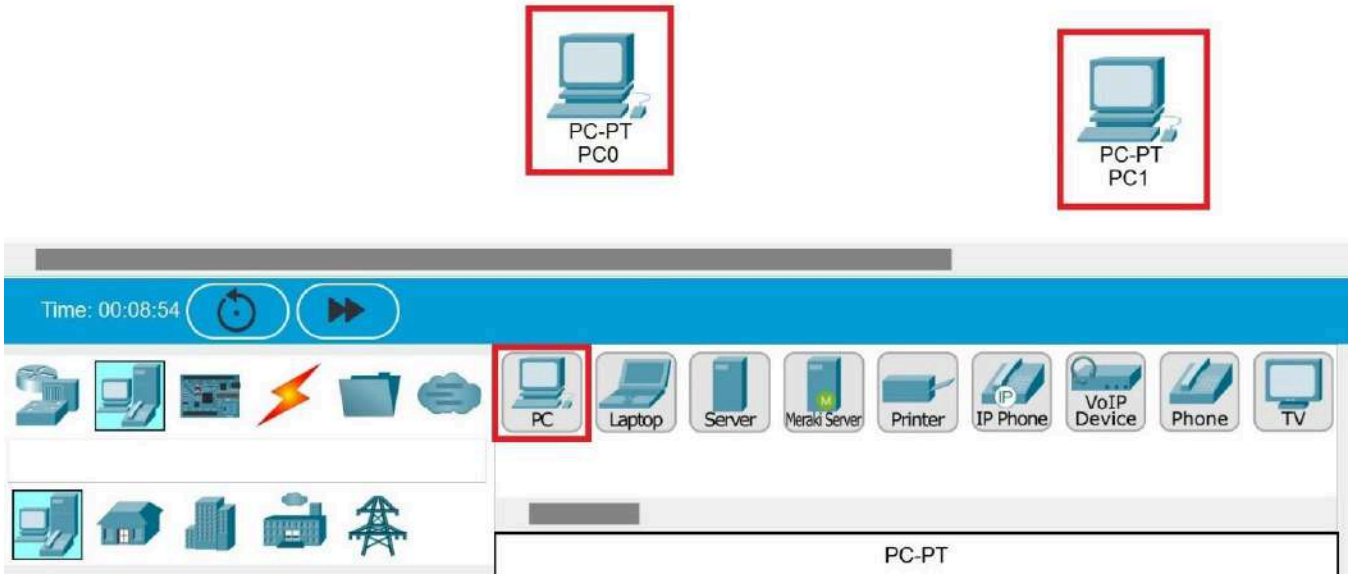
3. In the bottom left corner of the screen, locate the **toolbox area**.



4. In the **toolbox area**, select **End Devices** and then select **End Devices**.



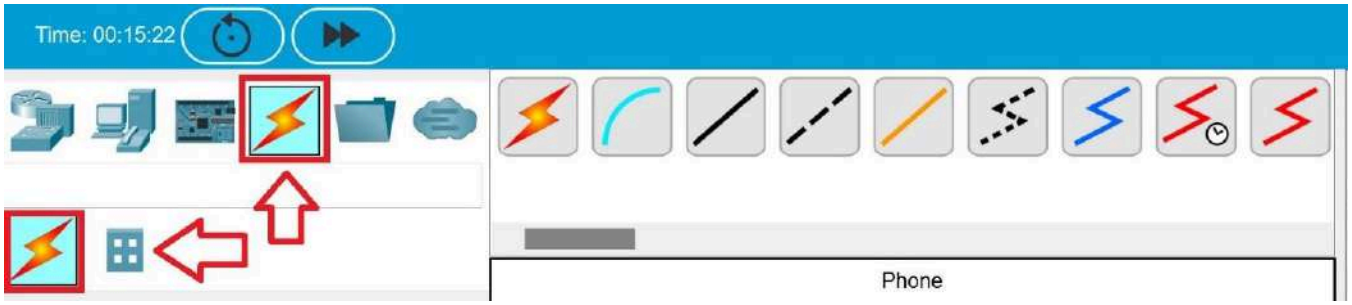
5. Drag two **PC** objects onto the logical diagram.



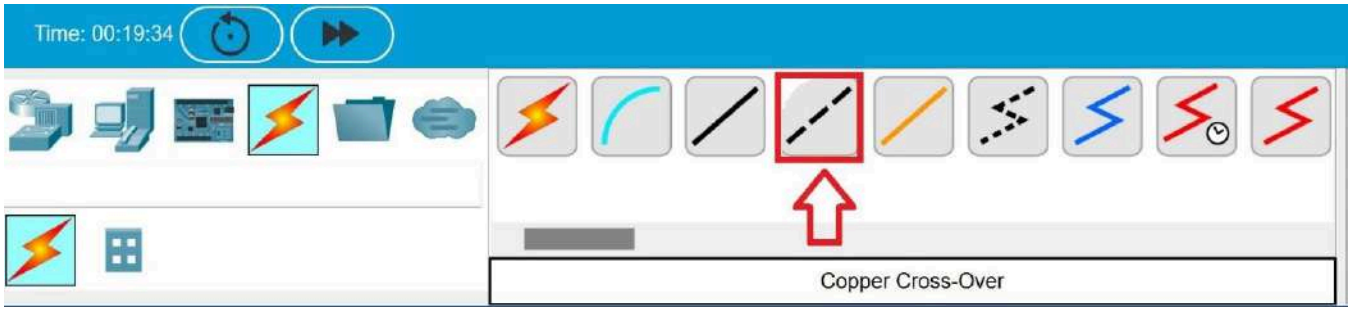
TASK B

Our two PCs each have a network card that we will configure in the next task when we set up the protocol. Before configuring the protocol, we need a medium to carry the signal between the devices. For this exercise, we're going to be using a wire. When one wire connects two devices directly, we need to use a special wire called a cross-over cable. In this task, we will connect the two devices using a copper cross-over.

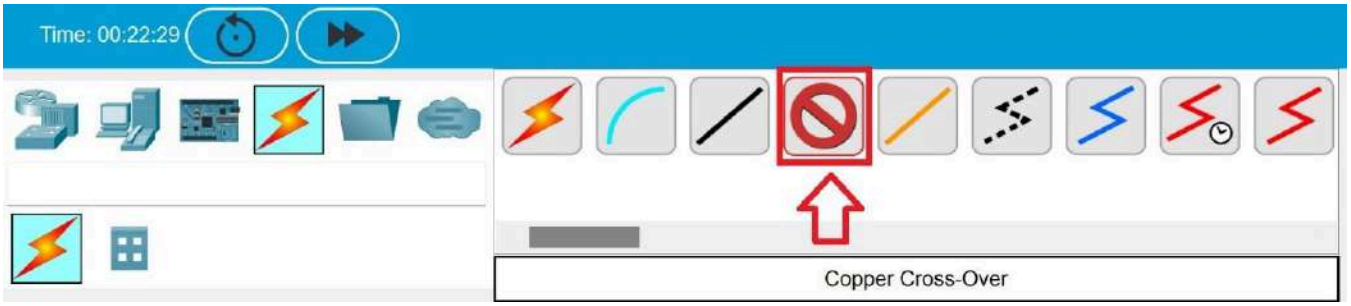
1. In the **toolbox area**, select **Connections** and then select **Connections**.



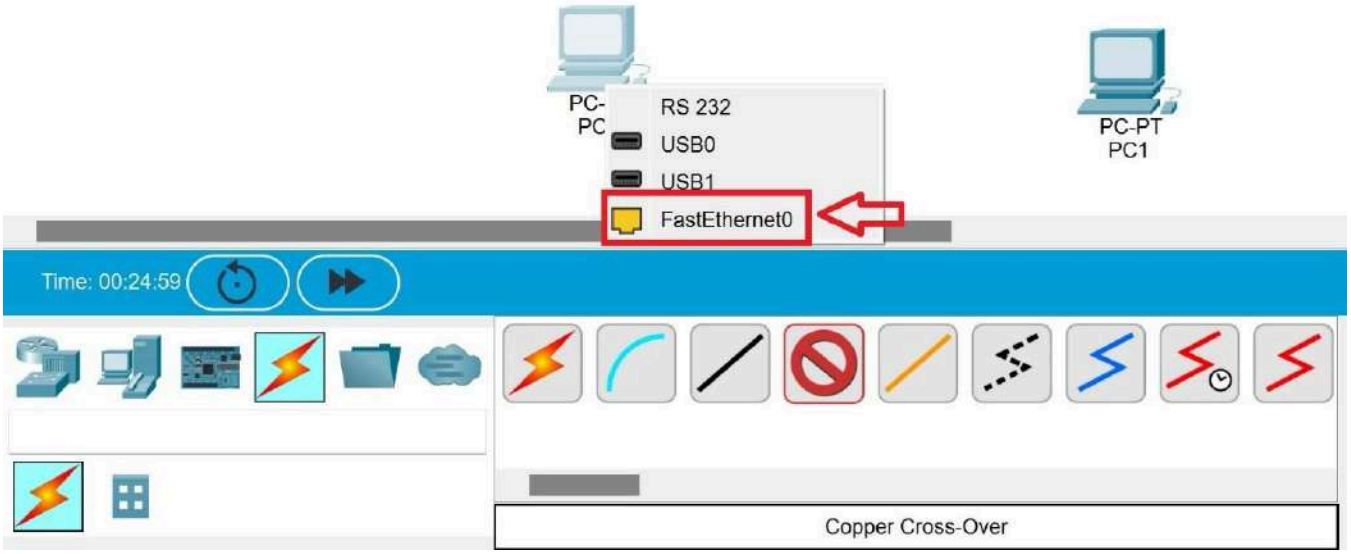
2. Locate the **Copper Cross-Over** cable button.



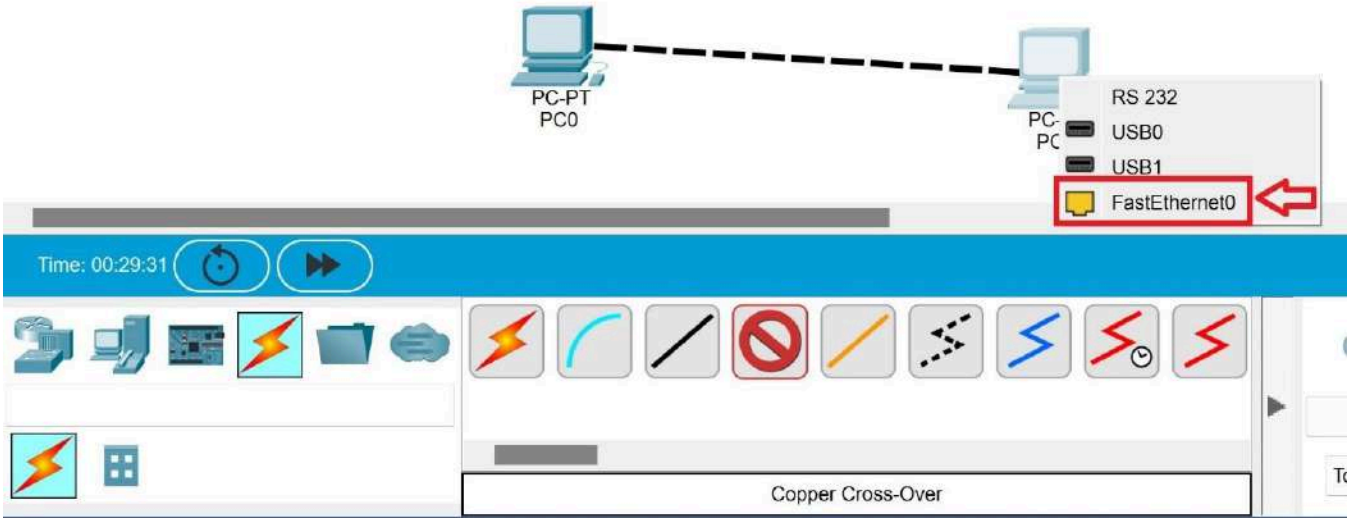
3. Click the **Copper Cross-Over** button (it will change to a red circle with a slash through it.)



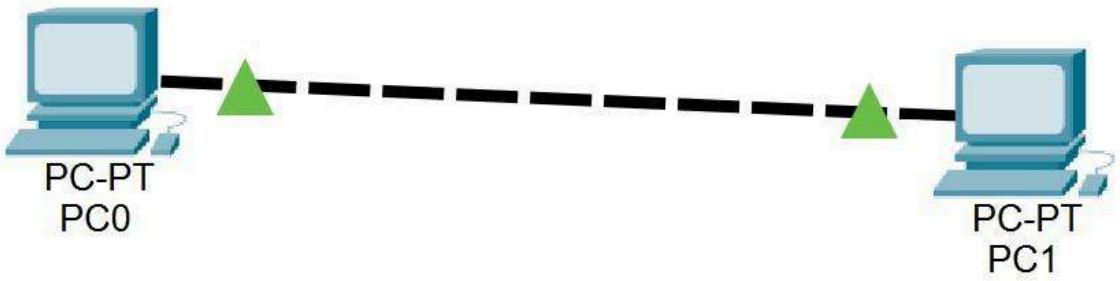
4. Click the **PC-PT PC0** device. A shortcut menu will appear. On the shortcut menu, click **FastEthernet0** to connect one end of the wire to the **PC0** network card.



5. Click the **PC-PT PC1** device. A shortcut menu will appear. On the shortcut menu, click **FastEthernet0** to connect the other end of the wire to the **PC1** network card.



6. Notice the link indicators on each side of the wire are green, indicating that the network card has detected a connection. (NOTE: If these were NOT green, it would indicate a problem.)



TASK C

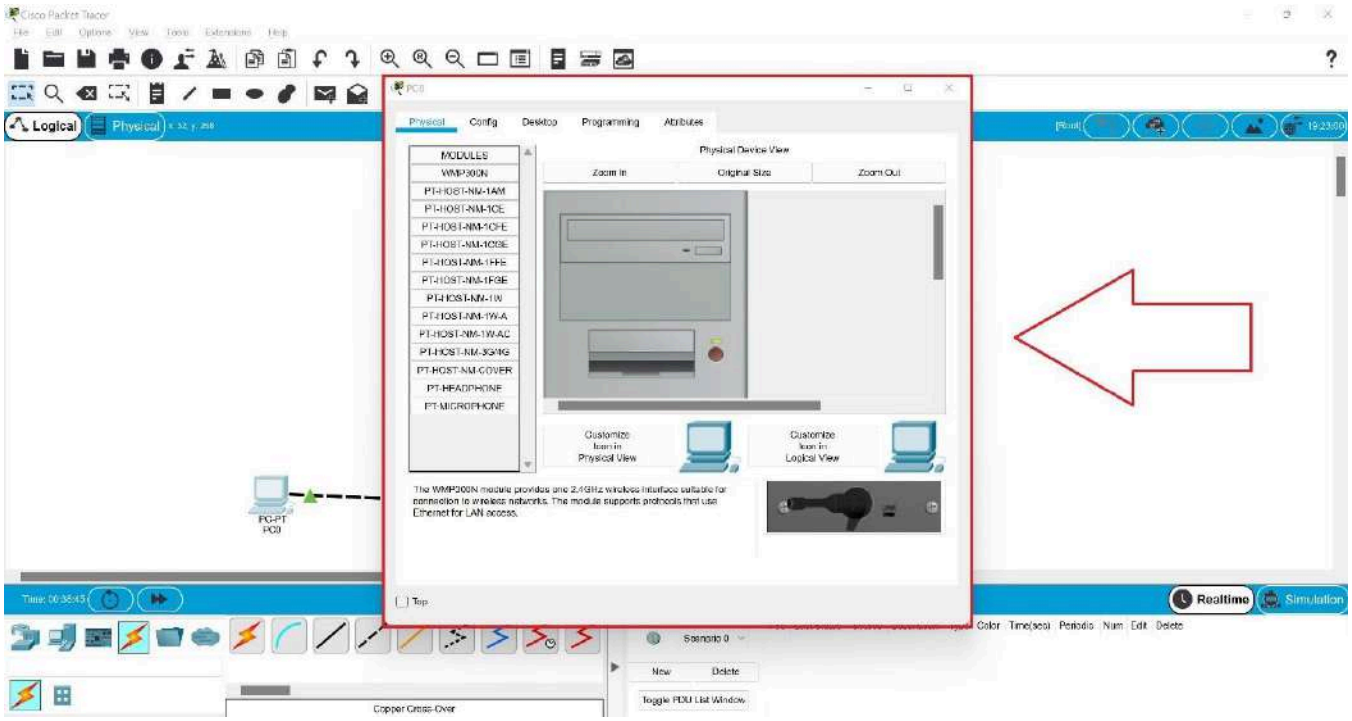
Now that the medium has been connected, we can verify that the computers support the same protocol and then test connectivity between the two nodes.

The PC object in Packet Tracer automatically supports the TCP/IP protocol. The two devices each need to be provided with an IP address and subnet mask in order for TCP/IP to work. (NOTE: We will discuss IP addresses and subnet masks in a later module.)

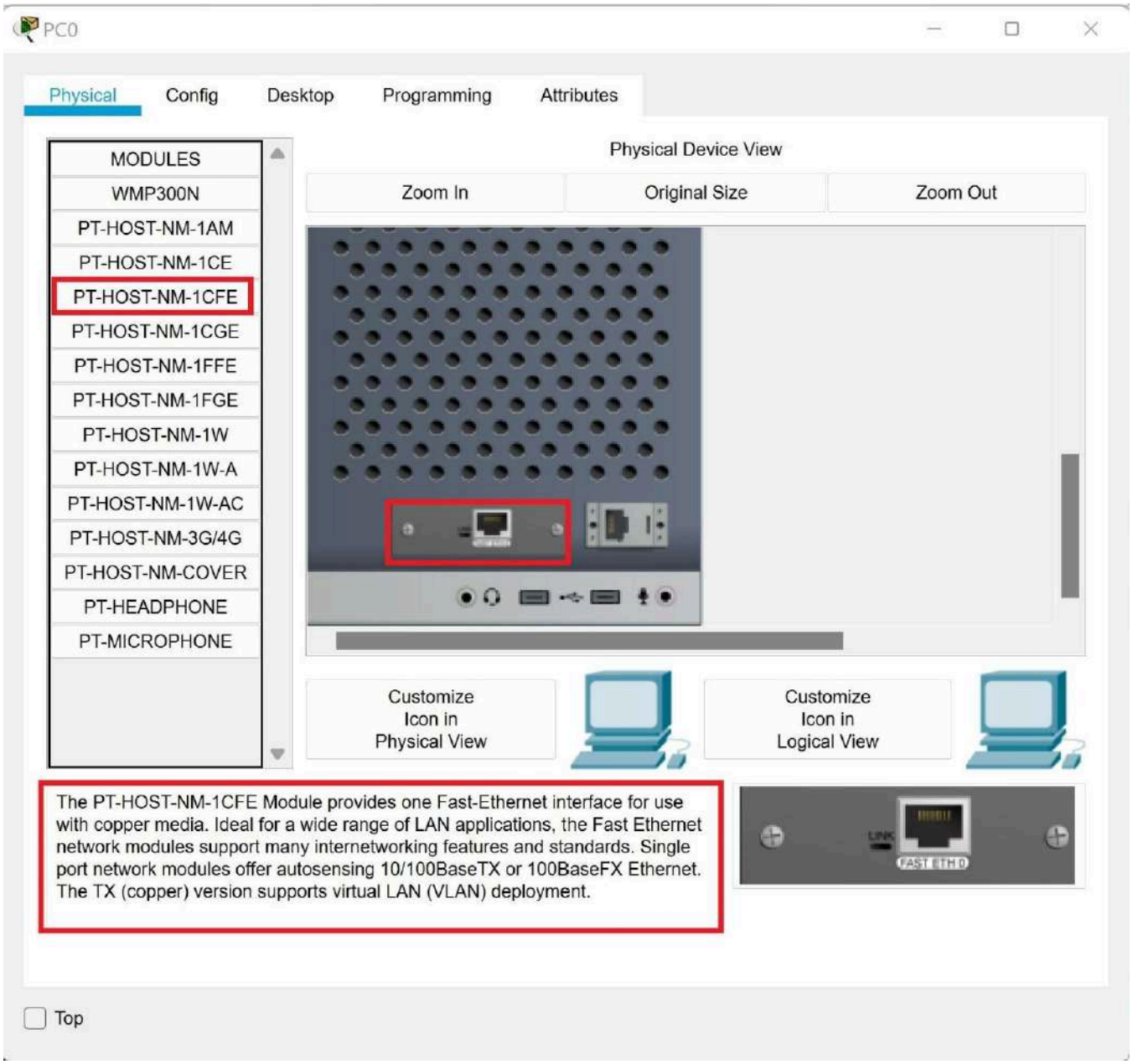
To test connectivity, network administrators commonly use the ping command. This command sends packets to a remote computer which then replies. A successful reply indicates round-trip connectivity.

In this task we will configure the protocol settings, and then we will test connectivity.

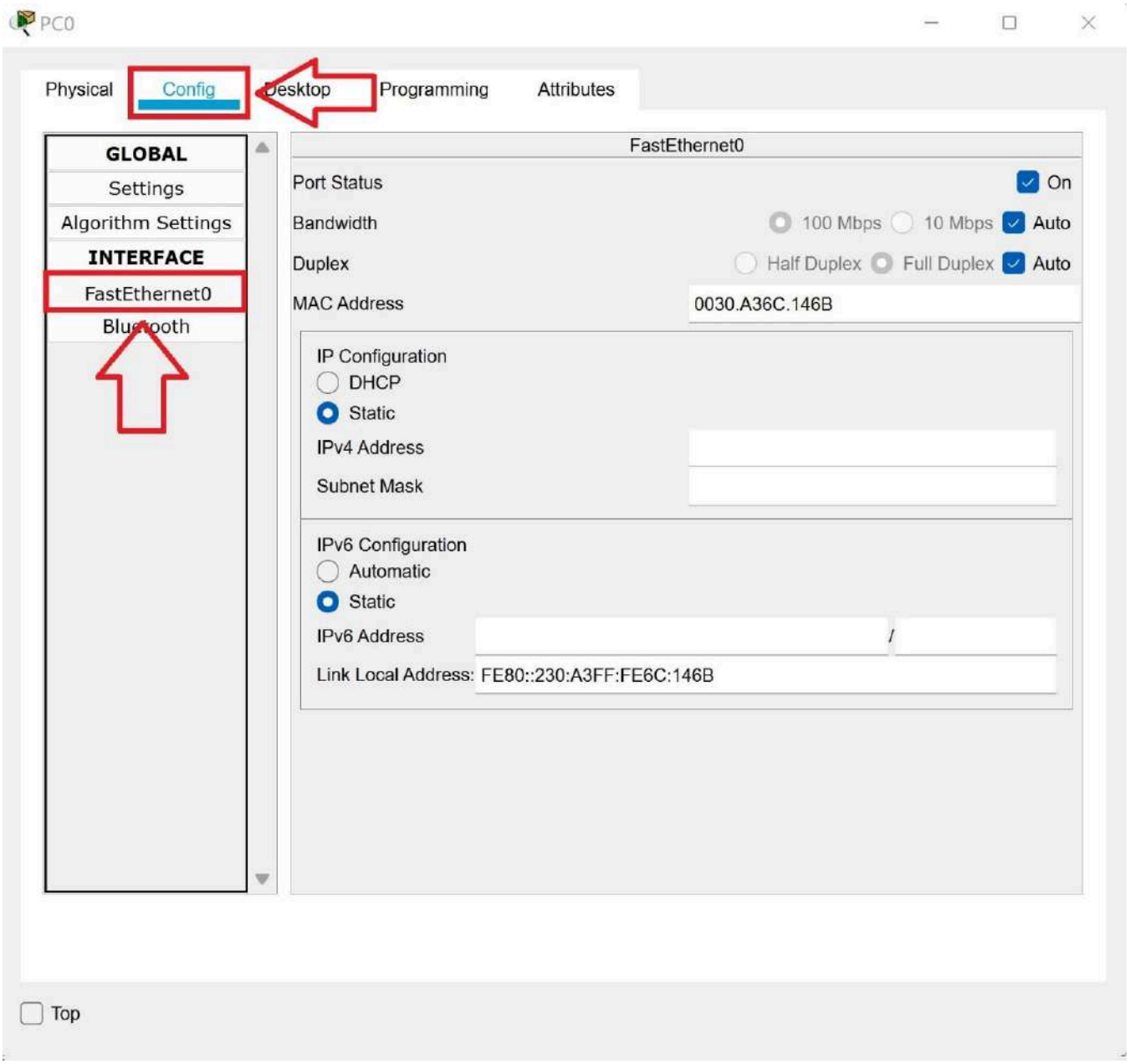
1. Click the **PC-PT PC0** device. The **PC0** dialog box will open.



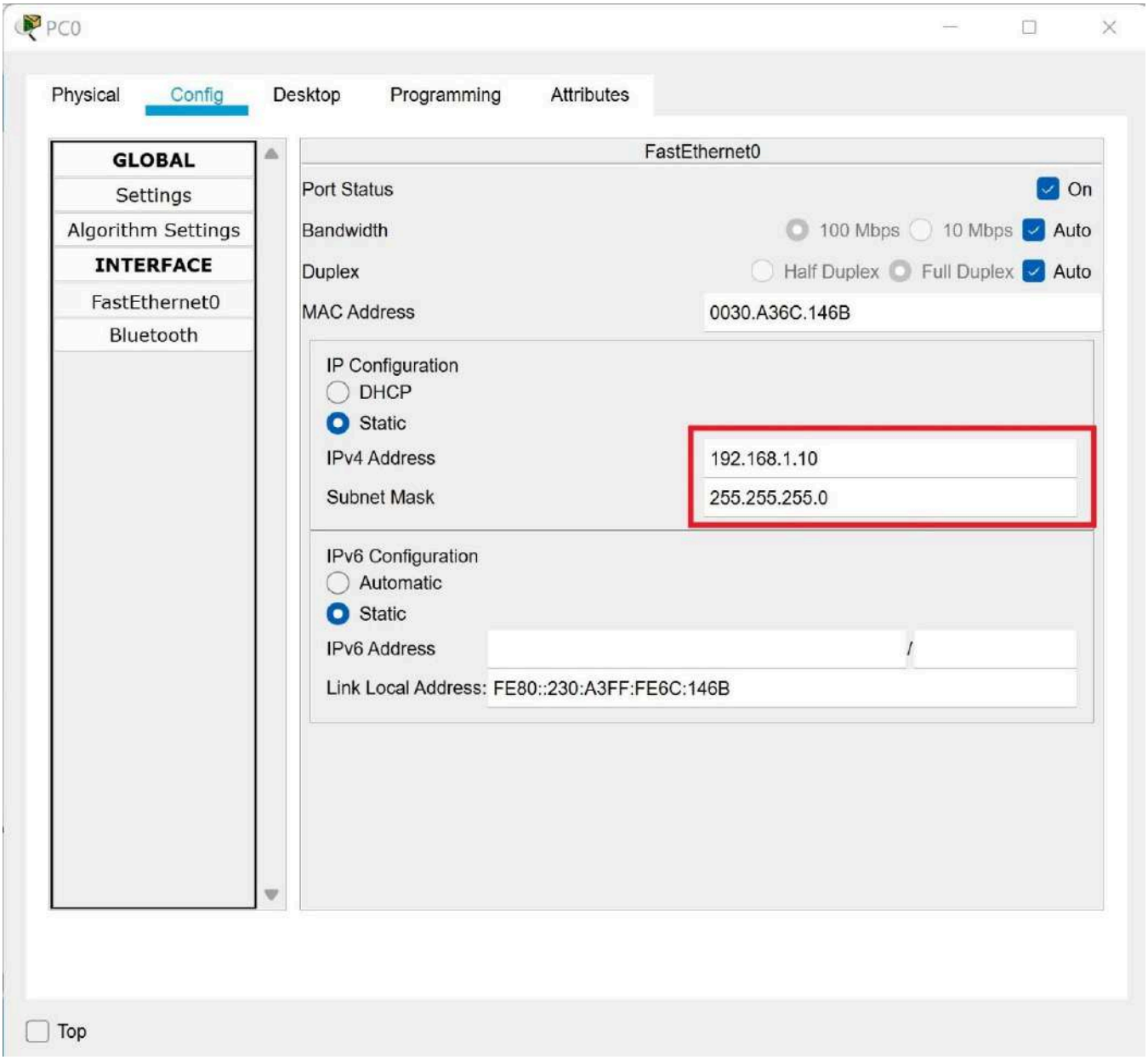
2. In the **Physical Device View** pane, scroll down to view the back of the **PC**. Notice the network card installed in the back of the computer. This is where the copper cross-over cable is connected. The equipment in Packet Tracer work exactly like the equipment would in real life.



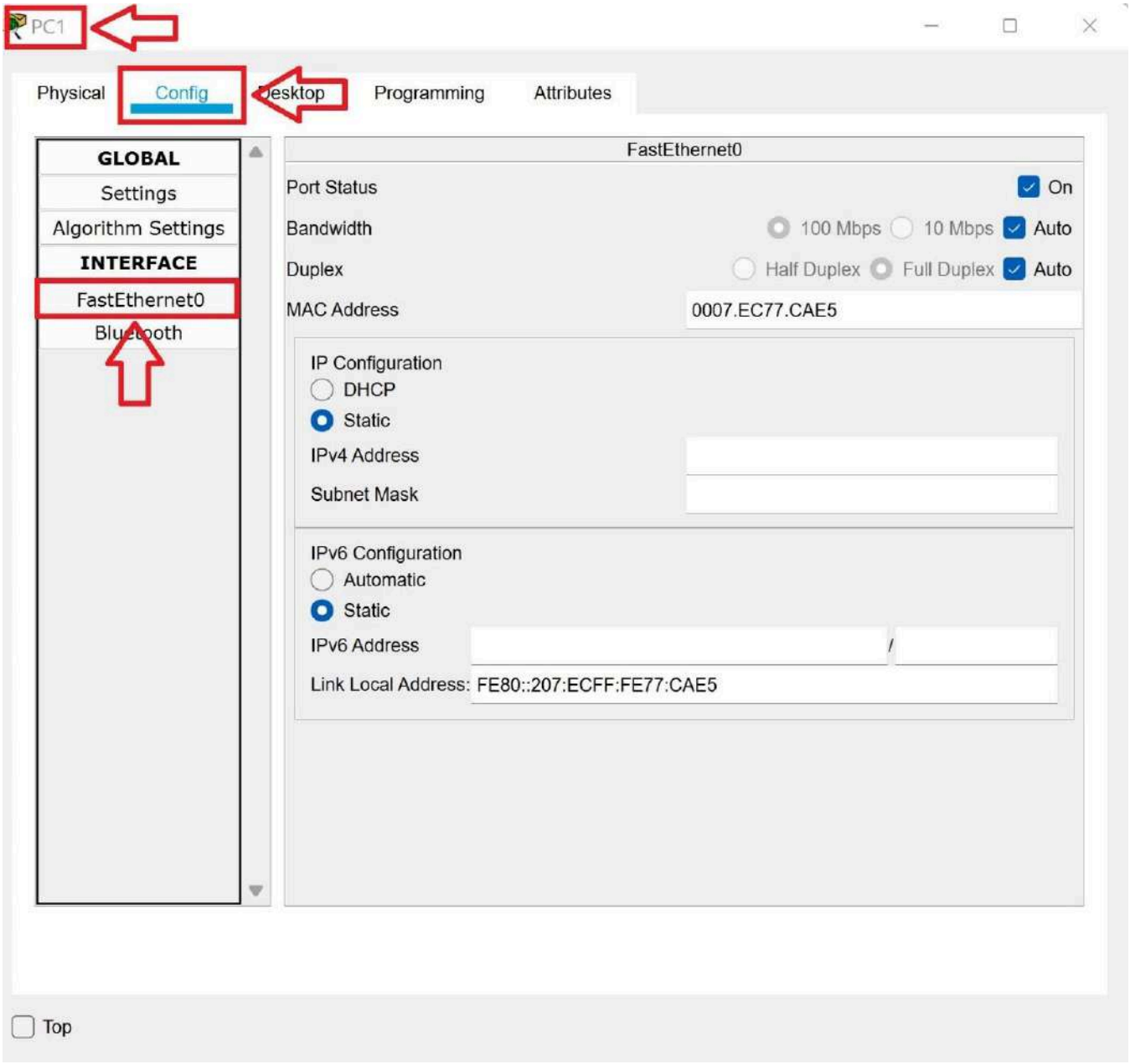
3. In the **PC0** dialog box, click the **Config** tab. Then in the **Interface** menu, click **FastEthernet0** to view the properties of the network interface card (NIC).



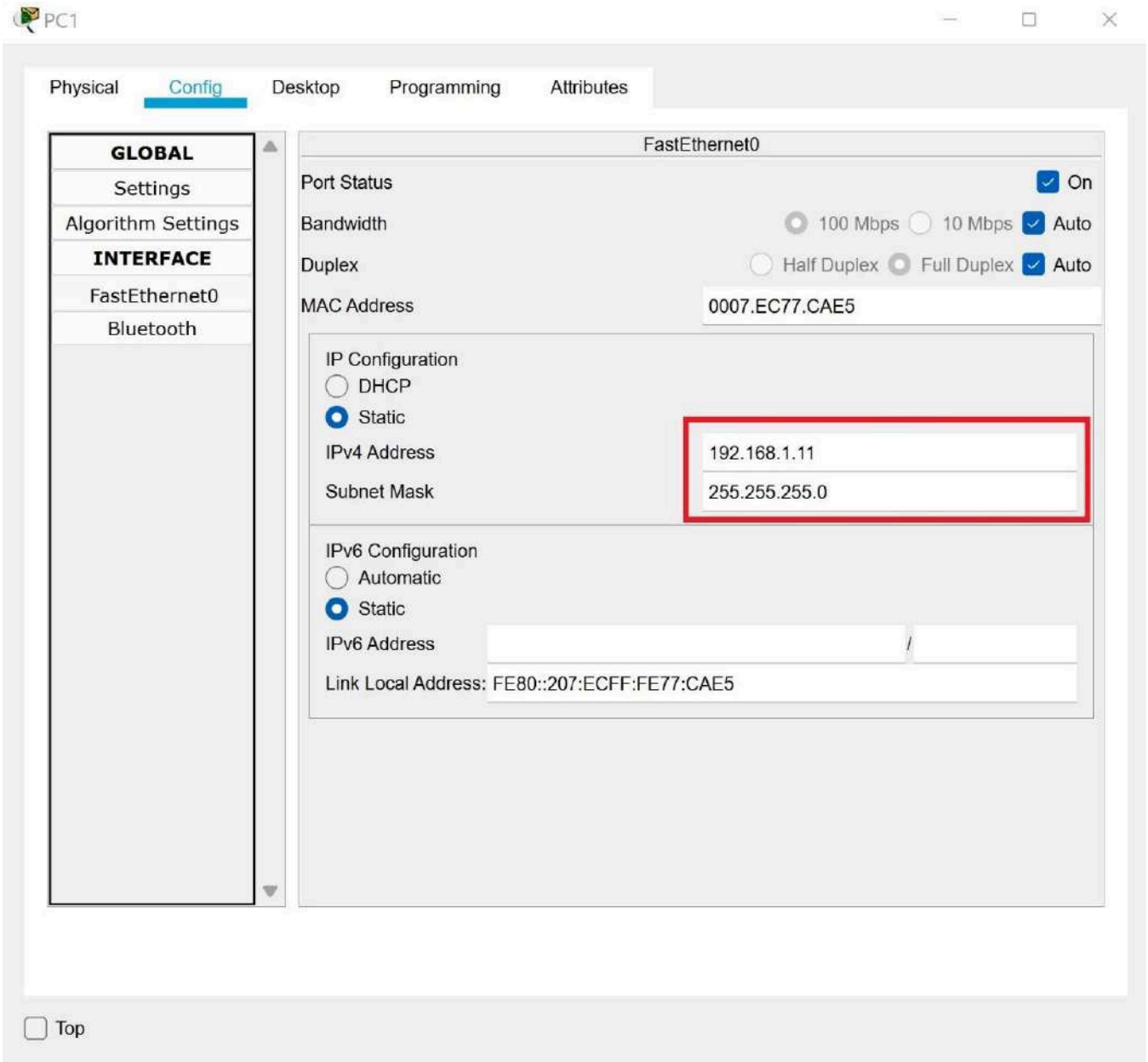
4. In the **IP Configuration** section, click in the **IPv4 Address** text box and enter a static address of **192.168.1.10**. Click in the **Subnet Mask** text box. The software will automatically enter a subnet mask of **255.255.255.0**.



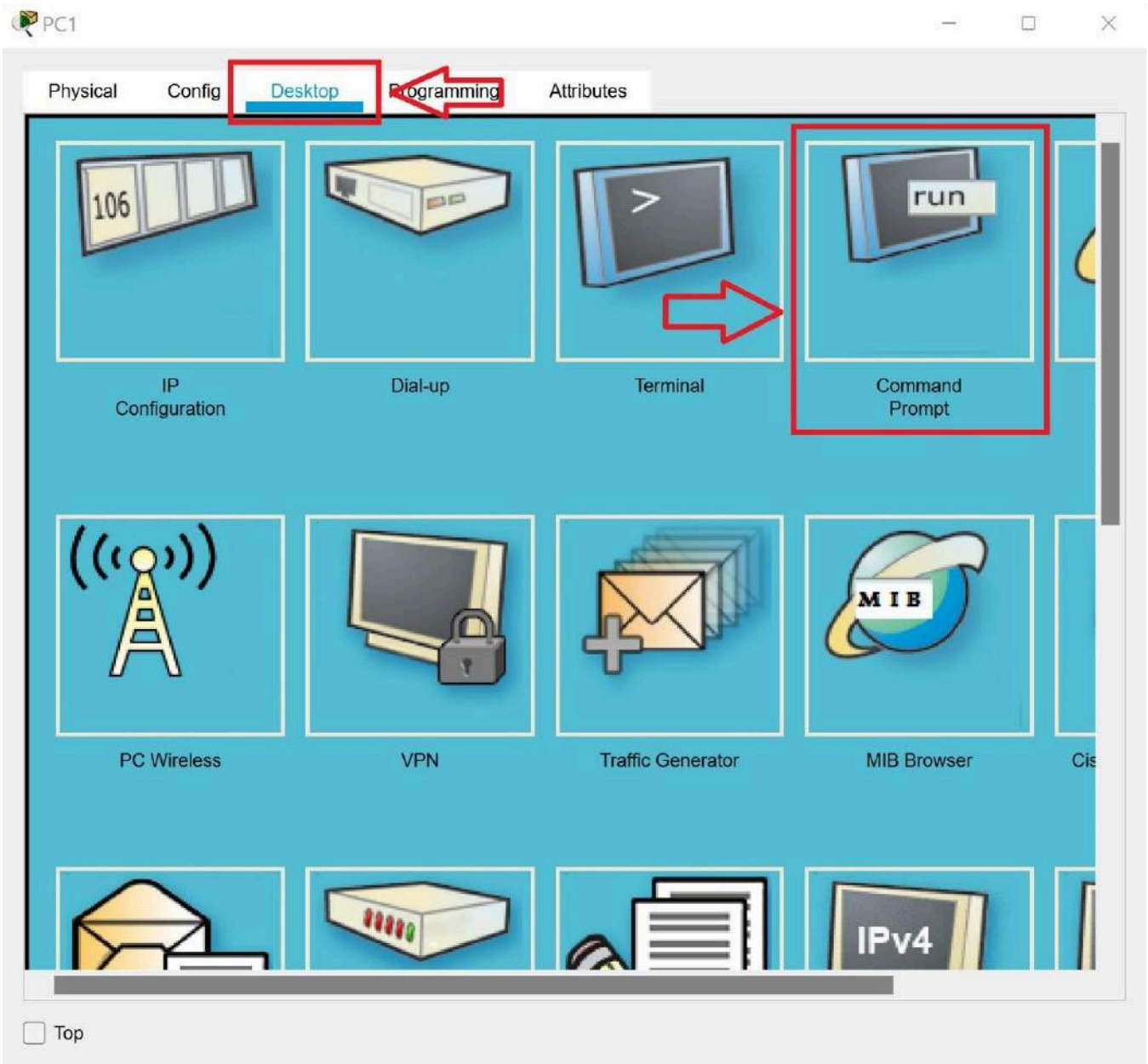
5. Close the **PC0** dialog box.
6. Click the **PC-PT PC1** device. The **PC1** dialog box will open. Click the **Config** tab. Then in the **Interface** menu, click **FastEthernet0** to view the properties of the network interface card (NIC).



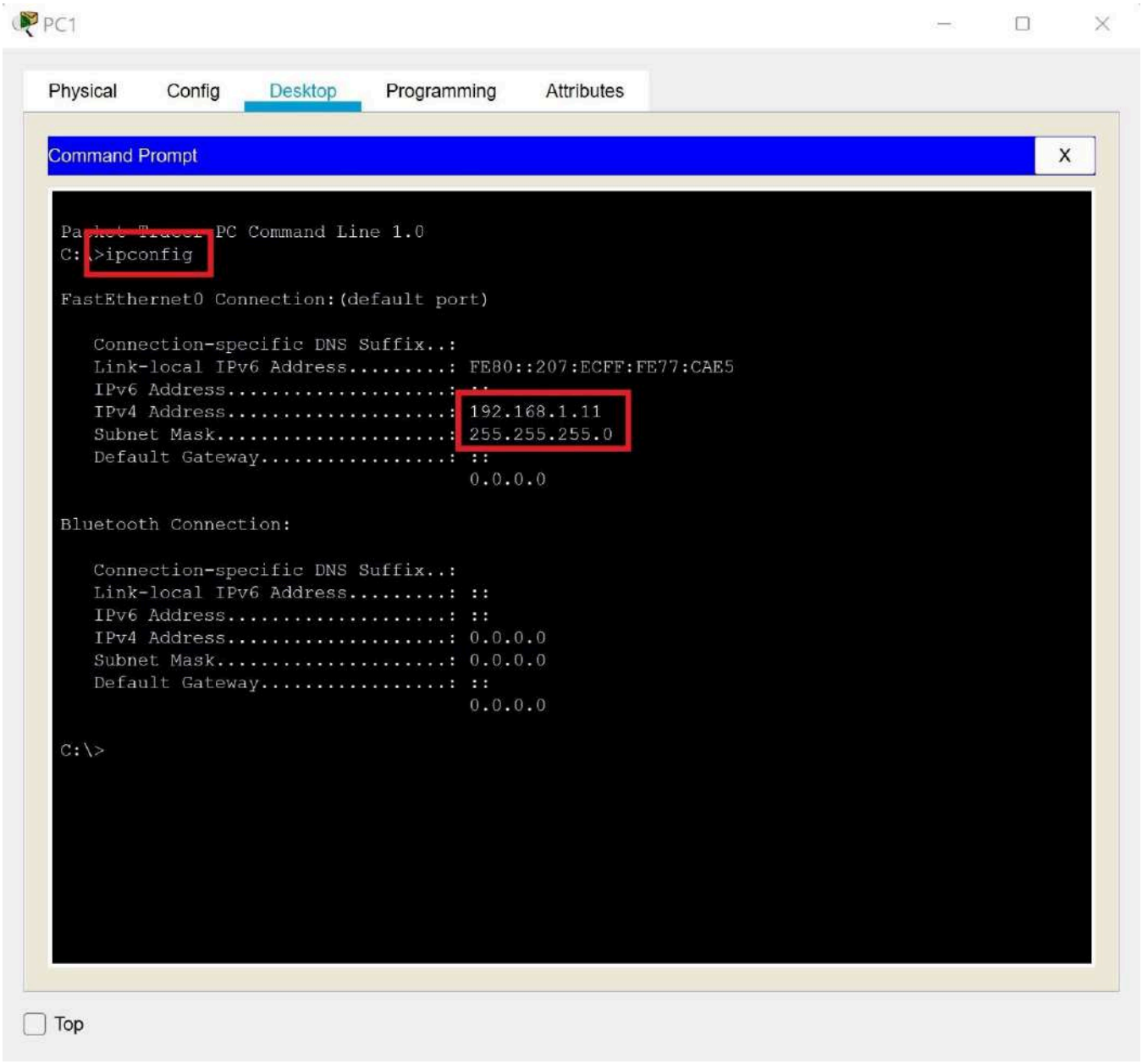
7. In the **IP Configuration** section, click in the **IPv4 Address** text box and enter a static address of **192.168.1.11**. Click in the **Subnet Mask** text box. The software will automatically enter a subnet mask of **255.255.255.0**.



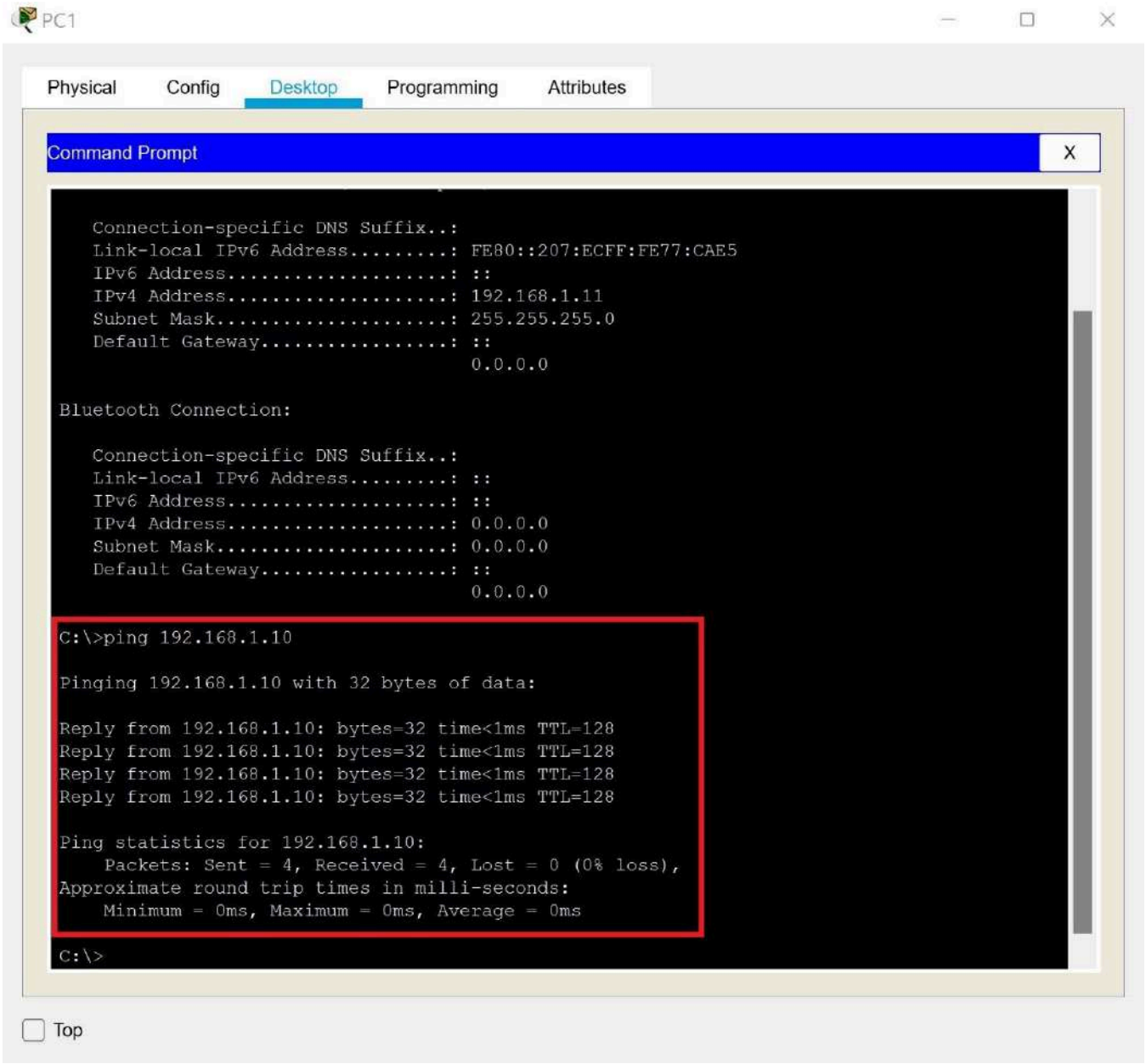
8. In the **PC1** dialog box, click the **Desktop** tab. Then, click the **Command Prompt** option.



9. In the **command prompt**, type **ipconfig** and then press the **enter** key on your keyboard to display the IP configuration of the computer. You should see the 192.168.1.11 address you entered in the Config tab.



10. In the **command prompt**, type **ping 192.168.1.10** and then press the **enter** key on your keyboard. You should see four replies come back from **PC0** verifying that connectivity is successful.



11. You may close the **PC1** dialogue box.
12. Congratulations! You created a simple network! The lab is over. You can close **Packet Tracer**. You do not need to save the file.

OSI Model and TCP/IP Model

Background of the OSI Model

In the 1970s, when companies started inventing protocols, most were proprietary. That means the hardware from different companies couldn't talk to each other. Technically, a network protocol is a set of rules for how data is transmitted. But for computers, they work like human languages. For two devices to "talk," they must support the same protocol. Throughout the 1970s and 1980s, there was debate about which protocol would be the best for networking.

The Open Systems Interconnect Model (OSI Model) started in the 1970s. It was adopted as a working model by the International Organization for Standardization (ISO) in the 1980s. The idea was to make a standard reference model that could be used to invent protocols that would be compatible. If all the vendors would agree to stick with one model, all devices would be able to connect.

Most countries wanted to adopt a protocol that matched the OSI Model. However, the United States, United Kingdom and France had been developing TCP/IP since the 1970s. It was released in 1981. TCP/IP was based on a different, four-layer model. Since TCP/IP was in use on the Internet, its popularity edged out OSI Model protocols.

Today the model is used for reference. But it still supplies a great description of how protocols work. You will often hear hardware talked about as a "Layer X Device." The layer is a layer from the OSI model.

The OSI Model is a seven-layer model. The layers are numbered from the bottom up.

7	Application
6	Presentation
5	Session
4	Transport
3	Network
2	Data Link
1	Physical

If you need a way to remember the OSI Model, there are two popular phrases. From the top down, All People Seem To Need Data Processing. From the bottom up, Please Do Not Throw Sausage Pizza Away.

How OSI Model works

We're going to be looking at following the data from the sending computer, down the protocol stack across the medium and then up the protocol stack on the receiving computer. It's important to remember that the OSI model is a theoretical model. So there's no actual protocol that works like this.

Application Layer

So on the sending computer, if it determines the data is intended for a recipient across the network, that's when it passes the data down to the application layer. The application layer, our application layer protocols, it's not the actual application itself. So if you're working in a browser, this doesn't kick in unless the browser realizes, I need to send this data across the network.

If I could only remember two words about the application layer, I tend to think ports and http. Now ports actually happened at a different layer. The reason I think that is that all the application layer protocols like https, DNS SMTP and that's why I think http to remind myself that it's all those family of protocols, they each have a port associated with them. And that port identifies an application or service on the machine.

In theory, it's going to add a header and a footer. It's coming from this sport is going to that court and then it passes the whole thing down to the presentation layer.

Presentation Layer

With the presentation layer, I tend to think of the sarcastic answer like presentation is to make presentable.
Famous activities at this layer will be compression and encryption

Certainly it's easier if it's not compressed but it does make things smaller, which saves space and can make things travel faster over the network. And encryption scrambles the data.

In theory it's going to add a header and a footer and pass the whole thing down to the session layer.

Session Layer

Its job is to start, stop and manage the session. In theory it's going to add a header and a footer and pass the whole thing down to the transport layer.

Transport Layer

So the transport layer is famous for error correction.

If you wanted to send a letter to somebody and you wanted to be absolutely sure they got it, you would take this down to the post office and you fill out a little green card with your name and address, and then they'd actually stick this green card to the letter. And then when the mail person would get to the recipients house, they'd ring the doorbell, that person have to come out sign the green card. The postal worker tears off the green card and that gets sent back to you when you get the green card back in the mail, you know that the person got your letter because they signed it saying they did. So when I met my wife I said look I travel 75% of the time, if we're going to live together do me a favor. If anybody ever rings the door for registered mail, whatever you do, don't sign the green card, it can't be anybody who knows me because everybody who knows me knows I'm not home and it can't be anything good. So let's say the lawyer writes up a letter dear shad, you owe us lots of money, love lawyer, they send it registered mail. My wife doesn't answer the door, she doesn't sign the card. What is the lawyer going to think? Well, at a minimum the lawyer has to think Shad didn't get my letter. And what are they going to do? What they're going to send it again and send it again and they're going to keep sending it until somebody signs that green card and they know I got it, that's TCP

TCP is famous for error correction. It works just like registered mail. So a TCP, the sender sends over a batch of packets, receiving computer gets the batch and it sends back an acknowledgement, yep I got the batch. If the sending computer doesn't get the acknowledgment after a while it just re sends the batch and it'll keep resending that batch until it gets the acknowledgement, just like registered mail.

TCP is known as reliable and connection oriented because we can guarantee with TCP that the data gets the other side.

UDP
UDP works like bulk mail, your favorite store sends out 5000 flyers, you don't call them up and go, hey dude, I got your flyer. I'm loving the fact that you have that great sale on Sunday. No, you just send it out best effort. UDP is unreliable, it's connection less. We use it in situations where speed is more important than reliability.

Think about the video you're watching right now. Videos are made up of tons and tons of pictures called frames. There's more frames in a video than the human eye can see. So what if a frame here or there doesn't make it to the other side. We wouldn't want the video to stop until everything came over perfectly. It's much more important that the video keep playing.

So at the transport layer, TCP connection oriented, reliable. It's like registered mail. UDP connection less unreliable, kind of like bulk mail.

In theory, the transport layer is going to add a header and a footer and the next thing it's going to do is pass it down to the network layer.

Network Layer

The network layer is responsible for logical addressing. Famous protocols at this layer would be an IP address. An IP address is a logical address. If I'm connected to a particular network, my computer gets an IP address. I disconnect from that network and go to a new network. I might, and I probably will, pick up a different IP address. That's why they call them logical addresses.

In theory, the network layer adds a header and a footer. With the IP address is coming from, the IP address is going to. At this layer, we talk about this whole entity as a packet. Everybody uses the word packet just to mean a chunk of data on the network, and that's totally fine.

Devices at this layer would be a router because routers look at the IP address. If they say a layer 3 device, they're talking about a router. Once the network layer adds the header and footer, it's going to pass it down to the data link layer.

Data Link Layer

The data link layer is responsible for physical addressing. A physical address would be a MAC address. MAC technically stands for media access control, but I think only trainers know that. Nobody talks about that. We just say MAC address. It's an address that's assigned to the network card by the manufacturer, and it never changes unless you're doing something strange. When they say physical address, they mean a MAC address.

At this layer, the data link layer is going to add a header and a footer coming from this MAC address going to that MAC address. Devices at this layer would be switches because switches look at MAC addresses.

Now if you're looking at this slide and thinking, wow, Shad, you have more headers and footers than you have data. Absolutely. Think about sending anything. You always end up with more packing material than you do whatever it is you're sending.

Now at this point, this thing that was a packet at the network layer is really big. You have some issues when you have really big data. It always reminds me of bad cellphone connections. I don't know if you've ever been on a bad cellphone connection. But on a good cellphone connection, I'm just chatting away to my wife, hi, honey. Yeah, I'll be home at six. I'll bring the milk. On a bad cellphone connection, I tend to do two things. One is rational, the other is irrational. The irrational thing I tend to do is yell. I'll be like, yes, home, six. There is absolutely no technical basis for the concept that if I raise my voice, it's going to get through a bad cellular connection better. But the rational thing I tend to do is I start to shorten my message. I'll be like, yes, home, six, milk. Because we all instinctively know that the smaller the message, the better the chance it has of getting through a poor communication medium.

This is a pretty big message right now. One of the things that happens at the data link layer is it's like a sushi chef with a California roll. It chops it up into smaller pieces, and those smaller pieces are known as frame. Whereas at the network layer we call it a packet. At the data link layer, we talk

about frames. The data link layer is going to add the header and footer from this MAC address to this MAC address, trap the whole thing up into frames, and hand it down to the physical layer.

Physical Layer

The physical layer is the network interface card, the NIC, and the medium itself. The physical layer, the network card on the resending computer, is going to receive all these frames and then it's going to send them out over the medium. It's going to come in on the physical layer on the receiving computer, come in the network card on the receiving computer. Network card on the receiving computer is going to gather it all up, reassemble all these frames, and then once it's got everything, it's going to pass it up to the data link layer on the receiving computer. Now we're going to follow this data back up the protocol stack on the receiving computer, but we're going to do that in the next video.

In this video of how the OSI model works. We're going to take a look at what happens on the receiving computer. So in the previous two videos, we followed the data from the application on the sending computer down through the application layer, down the protocol stack all the way to physical layer. The data came out the network card on the physical layer on the sending machine, crossed the medium, came in the physical layer on the receiving computer, which gathered it together all those frames together, reassembled them and passed the data up to the data link layer.

Now, we're going to follow the data up the protocol stack on the receiving computer. So the receiving computer at the data link layer reads the header and the footer that was put on by the data link layer at the sending computer. Computers get a lot of junk mail just like people. And so it's looking to see is this my Mac address or is it a broadcast Mac address? Because if it's not the data is not intended for me. And I could just dump it.

If it is intended for this particular computer it's my Mac address. It will strip off the header and the footer put on by the sending computer and pass the data up to the network layer.
So the network layer is going to read the destination IP address. Is this my IP address or is it a broadcast? Because if it's not it's not for me I can just dump it. If it is it strips off the header and the footer put on by the sending computer passes the whole thing up to the transport layer.

Transport layer sends an acknowledgement if necessary. Maybe it doesn't need to because it's UDP. Either way it's going to strip off the header and the footer put on by the sending computer passes the data up to the session layer. Session layer does its session thing, makes any needed adjustments figures out what session is going to on this computer, strips off the header and the footer put on by the sending computer, passes the data up to the presentation layer. Presentation layer, decrypts it, decompress is it makes any necessary translations, strips off the header and the footer put on by the sending computer. And then it passes the data up to the application layer.

The application layer reads the destination ports, figures out what actual application or service to give the data to on the receiving computer, strips off the header and the footer put on by the sending computer and passes the data to the receiving application. I think that is just the coolest thing, I could give this lecture a million times and I'm just as happy at the end of it as I was the first time I gave the lecture. I think it's so cool how the sending computer talks to the receiving computer and how each layer at the OSI model talks to the corresponding layer on the other side, through these headers and footers which that process is called encapsulation. I think it's really neat. It's important to know the layers don't talk to each other, but through encapsulation they talked to the layer on the other side. So that is how the OSI model works.

TCP/IP Model Layers

Transmission Control Protocol/Internet Protocol (TCP/IP) is a suite of protocols. TCP and IP are just two of the protocols in the suite.
TCP/IP was based on a four-layer model. It describes all the same functions as the OSI Model, just using less layers.
Here are the layers of the TCP/IP model and the protocols that make up the TCP/IP protocol suite.

Application	HTTP, HTTPS, SMTP, IMAP, POP, NFS, DNS, SNMP, DHCP, FTP, TFTP, Telnet
Transport	TCP, UDP
Internet	IP, ICMP, IGMP, ARP, RIP, OSPF, EIGRP, BGP, IPSec, NAT
Network Interface Layer	Ethernet (CSMA/CD, CSMA/CD), Token Ring, PPP, L2TP, PPTP

NOTE: We will discuss the names and functions of these protocols in later lessons.
Here is how the OSI Model relates to the TCP/IP Model:

<table>
<tr><th>OSI Model</th><th>TCP/IP Model</th></tr>
<tr><td>Application</td><td rowspan="3">Application</td></tr>
<tr><td>Presentation</td></tr>
<tr><td>Session</td></tr>
<tr><td>Transport</td><td>Transport</td></tr>
<tr><td>Network</td><td>Internet</td></tr>
<tr><td>Data Link</td><td rowspan="2">Network Interface</td></tr>
<tr><td>Physical</td></tr>
</table>

Data Encapsulation

Encapsulation is the process of adding information to the data at each layer. As the data is passed down the layers at the sending computer, delivery information gets added. If the information is added before the data, it's called a header. If it's added after the data, it's called a trailer.
At the receiving end, the data arrives in at the physical layer. As the data is passed up the protocol stack, the headers and footers are removed. This process is called de-encapsulation.
The important part to understand is that each layer uses this extra information to communicate with its counterpart. The layers do not communicate with each other. But, when you look at the raw data using a packet sniffer, you will see each of the individual layers.

Protocols

Web Page Protocols

HTTP

The HyperText Transport Protocol (HTTP) is used to send web pages across a network. It was developed by Tim Berners-Lee in 1989. The web pages are not encrypted.
HTTP is a stateless protocol. That means that the web servers don't retain any information about the web page after it's sent. If web applications need sessions, they must use another technology such as cookies.
HTTP resources are found using Uniform Resource Locators (URLs). The format of a URL is:
Protocol://hostname/filename
For example, consider the following URL:
https://www.akamai.com/solutions/security/ddos-protection
The protocol is HTTPS. The hostname is www.akamai.com. Historically, this means a computer named "www" in the akamai.com domain. The file name is "ddos-protection" in a folder named "security," which is in a folder named "solutions."
HTTP uses TCP port 80.

HTTPS

The HyperText Transport Protocol Secure (HTTPS) is an extension of HTTP that adds encryption. Originally, HTTPS used Secure Sockets Layer (SSL). SSL was developed by Netscape Communications in the early 1990s. After flaws were found in SSL, a new version was developed. Transport Layer Security (TLS) is replacing SSL.
The two protocols are not backwards compatible. That means that if a server uses TLS, but the client only supports SSL, communication can't happen. Web servers often run both protocols to make sure everyone can access the web page. Unfortunately, this can lead to more security problems.
HTTPS uses TCP port 443.

File Transfer Protocols

FTP

The File Transfer Protocol (FTP) is one of the oldest protocols used on the Internet to transfer files. Originally, many FTP shares were set up to support anonymous access. If you connected to an FTP site and it asked for a username, you could type in “anonymous.” You could supply any password and it would let you in. As more malicious users joined the Internet, FTP sites started to require authentication. But you may still find sites today that accept anonymous access.
As web-based file sharing has become more popular, FTP sites are not used as much. They are still often used to allow developers to upload changes to web pages.
FTP is not encrypted. If you use SSL or TLS to encrypt FTP, the protocol becomes FTP Secure (FTPS). If you use Secure Shell Protocol (SSH) to encrypt FTP, the protocol becomes SSH FTP (SFTP).
FTP usually uses TCP port 21.

TFTP

The Trivial File Transfer Protocol (TFTP) is an alternative to FTP. Trivial means small or insignificant. TFTP is usually used for small files, typically configuration files.
TFTP uses UDP port 69.

Email Protocols

SMTP

The Simple Mail Transport Protocol (SMTP) is used to send email. SMTP servers accept email from the users. They relay the email to other SMTP servers.
Originally, SMTP servers were configured as “open relays.” An open relay server accepts emails for other domains. They then relay the email to the right SMTP server. For example, suppose Akamai SMTP servers accepted email for Yahoo.com and then forward the email to Yahoo’s SMTP servers, they would be considered relays.
As people began to send spam, unsolicited email, relays caused a problem. Spammers could send the spam through the open relays. The relays would be blamed for the spam. No modern SMTP servers should be configured as open relays.
Ray Tomlinson, who invented email, sent the first email in 1971. At the time there was no good way to leave a message for someone as answering machines did not exist. Tomlinson wanted to make a system that would allow users to leave a message that could be collected by a specific individual on a computer. Tomlinson decided to use the “@” symbol to separate the recipient’s name from their location. The idea was that the user was “at” a location other than normal.
Email addresses use the format *alias@domain*.
SMTP is not encrypted. You can add SSL or TLS. Then the protocol becomes SMTP Secure (SMTPS).
SMTP uses TCP port 25.
The link to the article is for reference only. You do not need to read it.

IMAP

Internet Message Access Protocol (IMAP) is used to retrieve email. Messages are delivered to an SMTP server. The user connects to the server to get the messages. With IMAP, the messages are not copied to the user computer. That's why IMAP requires that you maintain a connection to the server while you're working with email. Email clients use either POP or IMAP, but webmail usually uses IMAP. The latest version of IMAP is IMAP4.
IMAP is not encrypted. You can add SSL or TLS. Then the protocol becomes IMAP over SSL (IMAPS).
IMAP uses TCP port 143.

POP

Post Office Protocol (POP) is also used to retrieve email. When the user connects to the server, the messages are downloaded to the user's computer. Then, the messages are typically removed from the server. The latest version of POP is POP3.
POP is not encrypted. You can add SSL or TLS. Then the protocol becomes Secure Post Office Protocol (POP3S).
POP uses TCP port 110.

Supporting Protocols

DHCP

Dynamic Host Configuration Protocol (DHCP) supplies IP addresses to clients.
For a client to become part of the network, it needs an IP address. The Network administrator can manually assign an IP address to a client device. This is called a static IP address. It's called static because it won't change until the administrator types in a new address. Static IP addresses are good when the device never changes networks. However, if the device moves, the administrator needs to type in a new address.
With DHCP, the client can be set to obtain a dynamic address from DHCP. The DHCP server is configured with a pool of addresses it can give out. When the client boots, it sends out a series of messages to find the DHCP server. The server gives it an address from the pool. The address comes with a lease. The lease is the longest amount of time the client can use the address. Before the lease expires, the client will contact the DHCP server to renew. If the DHCP server doesn't respond, when the lease expires, the client will need to find a new DHCP server.
DHCP makes it convenient for users to move between networks without having to program IP addresses. However, if no DHCP server is available, or if the DHCP server runs out of addresses, the clients can't work properly.

DNS

Devices use IP addresses to communicate with each other. However, it's not easy for people to remember IP addresses. The Domain Name System (DNS) is used to match a domain name to an IP address. When you use a domain name like Akamai.com to contact another device, DNS matches that to the IP address that the device needs to find the server.

DNS is a public database. It's also critical for networks to work. That makes DNS difficult to secure and a rich target for hackers.
DNS servers keep DNS records in database files called "zones." When DNS servers update each other's records, they use zone transfers.
DNS uses TCP port 53 for zone transfers. It uses UDP port 53 for client queries.

SNMP

Simple Network Management Protocol (SNMP) does exactly what it describes. It's a protocol for managing networks.
Devices that support SNMP have small databases called Management Information Bases (MIBs). The manufacturers name everything SNMP can monitor in the device in the MIB. Administrators set "traps" on the devices. The traps are thresholds that should trigger an alert. For example, support the network administrator wants an alert if more than 70% of the processor is being used on a server. They could set a trap for 70% processor usage.
When the trap is triggered, the SNMP Agent (client) on the device, sends the alert to a central monitoring service. Administrators then can get alerts from all the network devices in one place.
SNMP was not secure until SNMPv3. Only SNMPv3 should be used.
SNMP uses UDP port 161.

Week 2

Data Transmission and Media Access

Transmission Methods

Unicast Transmission (One-to-One)

Unicast transmission transmits data from a single source to a single destination. Unicast transmissions require the sending device to address the data to the receiving device. Any nodes that get the data, but are not involved in the transfer, ignore the data. Unicast transmission is the main mode used on LANs and the Internet.
Example:
HyperText Transfer Protocol (HTTP), Simple MailTransfer Protocol (SMTP), and File Transfer Protocol (FTP) all use unicast transmissions.
Broadcast Transmission (One-to-All)

Broadcast transmission transmits the data from a source to **all** the other nodes on a network. Data is usually sent to a special address called a broadcast address. All nodes understand that they should process data sent to the broadcast address. Nodes often use broadcast transmissions to advertise or find services on the network. Servers might advertise the service using broadcasts. If no advertisements have been sent, nodes broadcast a request for the service. If a server is present, it

responds to the request. Broadcasts are also used to find other devices or their addresses. Network services that rely on broadcasts generate lots of traffic.
Example:
Nodes that rely on DHCP to obtain an IP address send out broadcasts to find the DHCP server.

Multicast Transmission (One-to-Many)

Multicast transmission transmits the data to more than one device but not all of them. Multicast uses special multicast addresses. Nodes are predefined as members of a multicast group. Group members know to process data sent to the group address. Nodes that are not in the group ignore the data. Communication with nodes outside of a multicast group must be done through unicast or broadcast transmissions.
Example:
A video server transmitting video conferencing is an example of multicast transmission. Although more users might be using the service than are in the meeting, only the nodes in the meeting get the data from that meeting.

Ethernet

Now in human communication, if two people speak at the same time, the message often gets damaged. Imagine a large meeting everybody talking at the same time. If there isn't a method to organize who will speak and when, the meeting becomes chaos.

In networking because two or more devices share the medium. There must be a way to decide which device can transmit at any given time. More than one device transmitting at the same time destroys the data.

And so we have media access control which is the sarcastic answer. We are literally controlling who can access the media. Media is just the plural of medium at any given time. Early networks experimented with different ways of media access control.

For example, some of the early networks tried polling. In polling, there's a central device that checks whether each note has data to send. So with polling, each note has guaranteed access to the media, but if some of the notes don't have outgoing data, network time gets wasted.

Other early networks used tokens, and when I think of tokens I always think of talk shows where the host has a microphone. So whoever has the token can transmit the data and the token gets passed from note to note in a particular order. So again, we can control who has access to the medium but this system also can waste time by passing the token nodes that have no need to send data.

So ethernet was invented in the 1970s, it was adopted as a standard in the 1980s, but it's been updated and expanded continuously. It is the main method or most common method of media access control in modern networking.

And we're going to look at two flavors of ethernet, one will be carrier sense multiple access with collision detection (CSMA/CD). The other will be carrier sense multiple access with collision avoidance.(CSMA/CA)

CSMA/CD

In CSMA/CD, we have multiple nodes that are going to get access to the medium, which is a wired. CSMA/CD is just for wired networks, and what they do first is they listen to anyone transmitting. That's the carrier sense part of this. If nobody is transmitting, nope, looks clear, then they go ahead and transmit. Now, if two nodes happened to transmit at the same time, we had Carrier Sense Multiple Access just means that more than one computer can use the medium. If two devices transmit the same time so they both are saying, is anyone transmitting? They both decide, nope, it looks clear. They transmit at the same time there's a collision. In a collision, it's like, all the kids, so I get dinner, the data gets destroyed. Well, these two nodes are both going to detect the collision. They each set a random timer. Somebody's timer expires first and they start transmitting. When the other device's timer expires, the first one is already transmitting and so they have to wait for the first one to finish before it's their turn to transmit. So CSMA/CD, it's carrier sense. They listen to see if the media is free. Multiple access more than one device can use the media, but not at the same time. With collision detection, two devices transmit at the same time, there's a collision. They detected each at a random timer. Somebody's timer expires first, they start transmitting, the other one has to wait, and this is the most common media access control for ***wired*** networks.

CSMA/CA

This is the most common media access control for wireless networks. Wireless nodes, because they're wireless, can't detect the collisions.
Wireless nodes, because they're wireless, can't detect the collisions. What they have to do is avoid them. In CSMACA, nodes can transmit whenever they have data to send. But because they can't detect collisions, instead they send out a jam signal. It's essentially the electronic equivalent of saying, I'm going to transmit, so it's a packet that informs the other wireless devices that the node is about to start transmitting.
It waits a short time and then begins to send data. While that node is doing what it's doing, the other devices check for jam signals. If they see a jam signal, they will either stop transmitting or delay in transmitting. With CSMACA, it's just collision avoidance, and they avoid a collision by sending the jam signal.

Network Connectivity Devices

Network Interface Cards

NICs

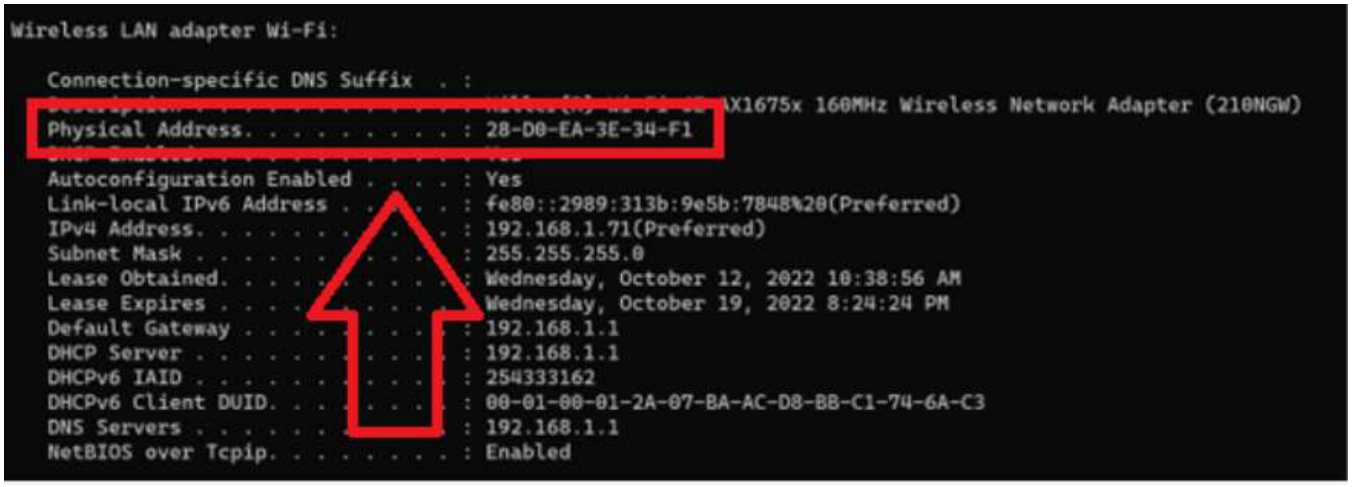
Network Interface Cards (NICs) connect devices to the network. Network adapter or network card are both alternate names for NICs. The NIC serves as an interface between a computer and the network. To connect to a network, a computer must have a NIC installed.
NICs can be built into the motherboard of the computer or can be connected using a port on the device. NICs can connect to either wired or wireless networks.

Duplex

Historically, NICs had to have their duplex set. The term duplex refers to how the network cards handle two-way communication. There were two settings for duplex: half-duplex or full duplex.
In half-duplex communication, the NIC can both send and receive. But it can't do both at the same time. NICs that are set to half-duplex function like a walkie talkie.
In full duplex, NICs can both send and receive at the same time.
The most important thing about duplex is that both devices need to be using the same setting. Imagine one device is set to half-duplex and the other is set to full duplex. The full duplex NIC can send and receive at the same time. Therefore, it will never stop transmitting. The half-duplex NIC expects that it will either be sending or receiving. Since the full duplex NIC on the other side never stops transmitting, the half-duplex NIC never gets a chance to transmit at all.
Modern network cards, and the devices they connect to, support auto-sensing. If the device on the other side requires half-duplex, they will select half-duplex. If the device on the other side supports full duplex, they will select full duplex. You should not have to adjust duplex in your career, but it is something you can check if two devices are having trouble communicating.

MAC Addresses

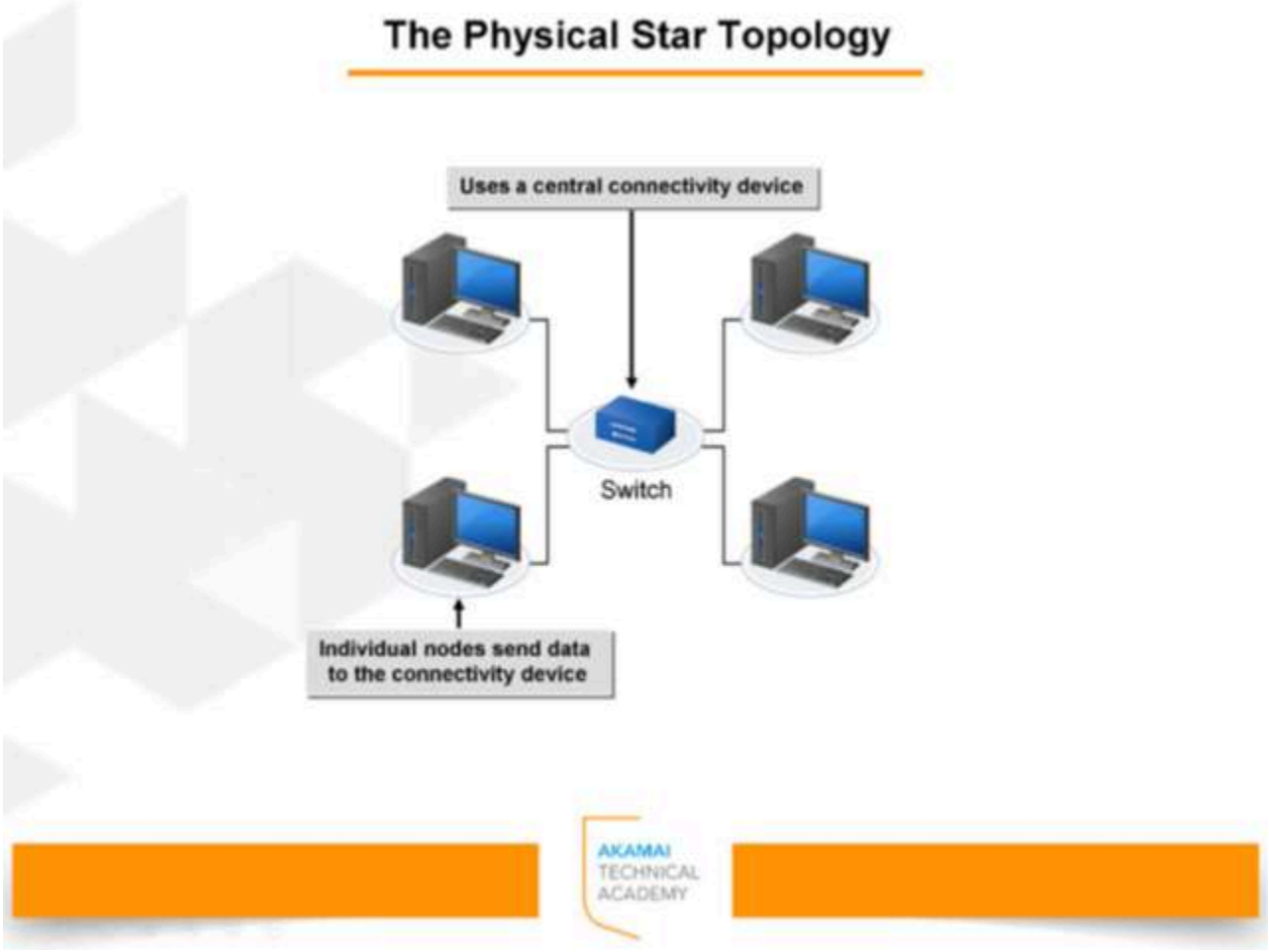
To deliver something like mail or data, the recipient must have a unique address. Imagine if there were two houses that had the same address. How would the mail system know where to deliver each letter or package?
The same is true for NICs. Each NIC must have a unique address. That address is called a Media Access Control or MAC address. It may also be called a physical address. The MAC address is a unique, hardware address assigned to the NIC by the manufacturer.
MAC addresses are 48 bits long. MAC addresses have six sets of two-digit hexadecimal numbers. The first three sets identify the manufacturer, and the last three sets identify that particular NIC.



Network Interface Cards

Hubs

It’s possible to connect two devices with a wire (or wireless) like you did in the Network Theory lab. However, networks usually have a lot more than two devices. In Ethernet networks, the network typically uses a central device to connect all the nodes. This redistribution point takes the data coming in and sends it to the receiving nodes. When all the nodes are connected to a central device, this is known as a star physical topology.



Early networks used devices called hubs. Hubs are also known as repeaters. That’s because these Layer 1 devices take the incoming signals and send it to all the ports on the hub.
The only problem with hubs is caused by the very nature of how they work. If a node sends data to the hub, it repeats the data to all the ports. That means that if any other node was about to transmit, there will be a collision. Then both nodes will have to wait for a random time delay. The more devices connected to the hub, the more collisions the hub will have. The more collisions on the network, the slower the network runs. “Collision domain” is the term that describes all the nodes who can create a collision with each other. When you use a hub, all the devices are in one big collision domain.
Modern networks don’t use hubs, they use switches.

Switches

Switches can also receive incoming data and send it to other nodes. When the switch first turns on, it acts like a hub. It sends all the data to all the nodes. This is called “flooding” the data.
To properly address data, the sending node must find the receiving node’s MAC address. Typically, the sending node has only the IP address of the receiving node. To find the MAC address of the NIC with a particular IP address, nodes use a protocol called Address Resolution Protocol (ARP).
To resolve the receiving node’s IP address to its MAC address, the sending computer sends out an ARP broadcast. Suppose the sending computer needed to know the MAC address of a receiving computer with an IP address of 192.168.1.10. It would send an ARP broadcast, “192.168.1.10 what is your MAC address?” The switch sends all broadcasts to all ports. If 192.168.1.10 is on the network, the ARP broadcast reaches the device. It responds by providing the sending device with its MAC address.
As ARP broadcasts go through the switch, the switch makes a note of which MAC address(es) are on each port. The switch stores this information in its Content Addressable Memory (CAM) table. When data comes in, the switch looks at the destination MAC address. If the CAM table lists a port for that MAC address, the switch sends the data just to that one port. Because switches send data based on the MAC address, they are Layer 2 devices.
Because switches send the data to just the one port with the receiving node, that is the only device that could have a collision with the data. Therefore, each port on the switch is a separate collision domain. Replacing a hub, where all the ports are one big collision domain, with a switch, where each port is a collision domain, can really speed up a network.

Managed Switches

Managed switches have firmware. The firmware functions as an operating system that can be used to program the switch with security features.

Packet Sniffers

Packet Sniffers allow administrators to capture network traffic. Then the administrator can examine the actual data passing across the network.
To capture traffic, the switch needs to send the data to the packet sniffer. However, the switch will only send data to the packet sniffer if the sniffer’s MAC address is listed as the receiving node.
To allow packet sniffers to collect all the data on a switch, administrators must configure port mirroring on the switch. This tells the switch to copy (mirror) all the data passing through the switch to one port.
By default, NICs ignore data that is not either a broadcast or addressed to their MAC address. When administrators install a packet sniffer on a computer, they must tell the NIC to process all the incoming data even if it’s not a broadcast or addressed to the node’s NIC. They do this by putting the NIC into promiscuous mode. In promiscuous mode, the NIC sends all the data up the protocol stack to the packet sniffer.

Routers

Technically, any device that is connected to two or more different networks, and can pass information between them, is a router. Routers connect multiple networks that use the same protocol. Routers only work with routable protocols. Routable protocols assign an address to the network and to each node on the network. TCP/IP is a routable protocol. With IP addresses, part of the IP address is the network address. The remaining part is the node address.

All devices that support TCP/IP have a routing table. In a node that isn't a router, the routing table lists the address of the local network. It also lists the default gateway, the address of the local router. The device uses the routing table to make routing decisions. Data that's destined for the local network is sent directly to the destination device.

When data comes in that's destined for a different network, nodes send the data to the default gateway. The router uses the network address portion of the destination IP to decide what to do with the data. If that router isn't directly connected to the destination network, it sends the data to another router. The data is delivered when it finally reaches a router connected to the destination network.

Routers have more entries than nodes in their routing tables. By default, every device lists the local network in their routing tables. Routers exchange their routing tables with other routers. In that way, routers "learn" about other networks. Then they can forward data to remote networks.

When a broadcast comes into a network card on a router, the router knows that the broadcast was intended for all the nodes on that network. Broadcasts are not intended for nodes on other networks. That is why routers do not forward broadcasts. A broadcast domain is composed of all the nodes on one network. Routers separate broadcast domains.

A router can be a dedicated device, incorporated into a multi- function device, or can be implemented as software. Even a regular computer, with two NICs, can be configured as a router. Typically, when professionals use the term router, they're talking about a dedicated device.

Wireless Networks

WLAN Basics

A Wireless LAN (WLAN) is a self-contained network of two or more computers connected using a wireless connection. A WLAN spans a small area, such as a small building, floor, or room. A typical WLAN consists of client systems such as a desktop, laptop, or personal digital assistant (PDA) and wireless connectivity devices such as access points. The access points interconnect these client systems in a wireless mode or can connect to a wired network. WLANs allow users to connect to the local network or the Internet, even on the move.

At a minimum, a WLAN needs a Wireless Access Point (WAP). If you see the term "access point" it typically refers to a WAP. Historically, WAPs functioned as a bridge between the wireless clients and the existing wired network. Modern wireless networks, especially SOHO networks, may not connect to a wired network. However, most WAPs are multifunction devices that do more than connect wireless devices together. Most are routers, DHCP servers and may include security software like a firewall.

Association

Wireless clients don't "connect" to wireless networks. Instead of "connect," wireless clients "associate" to the wireless network.

To associate to a wireless network, clients must know the Service Set Identifier (SSID). From the experience of a user, the SSID is the name of the wireless network. However, the SSID functions like a password. If the client doesn't know the SSID, it can't associate.

Most WAPs broadcast their SSID using beacon frames. They have information about the communication process, such as the SSID, channel number, and security protocol information. This allows end users to click on a list of available wireless networks and then connect to the right network.

For rudimentary security, administrators can turn off the broadcast of the SSID. Then the network will appear in the list as a "hidden network." Users will be prompted to enter the SSID before they can connect. This isn't very good security because there are multiple ways a hacker can find out the SSID even if the broadcast is turned off.

Protection

If all a user needs to know to associate to the wireless network is the SSID, the network is considered an "open" network. Open networks available to the public are also called hotspots. Open networks do not have any form of encryption. Data sent across an open network can be seen by anyone with a packet sniffer. That is why when users connect to open wireless networks, it's recommended to add a VPN connection. Virtual Private Network (VPN) connections are typically used for remote access (someone who is outside of the company getting access from remote), but they are always encrypted.

Open networks usually have a captive portal. A captive portal is a web page that opens when the client connects to the wireless network. It usually will have a disclaimer and ask the user to agree to behave legally while connected to the network. If the network requires users to log in to use the network, it will have the login dialog box.

Networks that are protected by encryption will either prompt the user to enter a password or to login. Just knowing the SSID isn't enough to log in to a wireless network that uses encryption.

802.11 Standards

The 802.11 standard is a family of specifications developed by the IEEE (Institute of Electrical and Electronics Engineers) for wireless LAN technology. Whenever you see "802.11," you should know that the topic is Wi-Fi or wireless networking.

Standard	Year	Speed (Mbps)	Frequency (GHz)	Range (Meters)	Features
802.11a	1999	54	5	20	
802.11b	1999	11	2.4	100	
802.11g	2003	54	2.4	100	
802.11n	2009	600	2.4/5	70	MIMO

802.11ac	2013	6933	2.4/5	100	MU-MIMO
802.11ax	2021	9608	2.4/5/6	240	OFDMA

802.11a uses the 5 GHz frequency. At the time 802.11a was approved, there were relatively few devices that used this frequency. However, higher frequencies have taller radio waves. These taller waves don't travel as far. They're also more easily blocked by solid objects like walls. Thus, the change to the 2.4 GHz frequency for 802.11b.
Starting with 802.11n, Wi-Fi devices support a technology called multiple-input multiple-output (MIMO). With MIMO, signals are sent via multiple paths at the same time. By sending the data via multiple paths, if one path is blocked, the data still arrives.
802.11ac (Wi-Fi 5) introduced Multi-User MIMO (MU-MIMO). MU-MIMO works like MIMO but manages multiple devices more efficiently.
802.11ax (Wi-Fi 6) improved on how signals could be sent by using Orthogonal frequency-division multiple access (OFDMA). OFDMA is a multi-user version of the orthogonal frequency-division multiplexing (OFDM) digital modulation scheme. OFDMA achieves multiple access by assigning subsets of subcarriers to individual users. This allows several users to send simultaneous low-data-rate transmissions. That means modern wireless networks, which have more devices than ever before, using Wi-Fi 6 can manage the increase in devices without sacrificing speed or distance.
Do note that for wireless devices to communicate, they must operate at the same frequency. Therefore, an 802.11a device would not be able to communicate with an 802.11b or 802.11g device.

Wi-Fi Modes

Wireless devices can work in two modes: infrastructure and ad hoc. Infrastructure networks use a centralized device (WAP) to send data between the nodes. Ad hoc networks allow wireless devices to communicate directly with each other without a central device. Ad hoc networks are never encrypted. Historically, ad hoc networks were popular in the early 2000s. They were used to connect hardware like printers when users did not have a wireless access point. Now, they are often used to program Internet of Things (IoT) devices like wireless security cameras, light bulbs and thermostats. These ad hoc networks do not present a security issue since they're used for just long enough to send the information about the WAP to the IoT device.

WPS

Wi-Fi Protected Setup (WPS) is a feature that many wireless NICs and WAPs support. The idea was to simplify connecting to the wireless network. Users could push a button on the WAP and a button on the NIC and they would sync up. This way, the user did not have to enter an SSID or an encryption key to join the wireless network. However, WPS uses a short PIN (four-digit number) that can easily be cracked.

Wireless Security

Basic Wireless Security

There are three basic things you can do to secure a wireless network.
First, try to make sure the wireless network is only available in the areas where you want to supply wireless coverage. If the wireless network extends too far, attackers can try to hack the network without being seen. For example, suppose you wanted to supply wireless coverage for an entire building. If the wireless network

extends to a street behind the building, attackers can sit in a car and try to hack the network from the street. In that case, most WAPs allow you to decrease the transmission power so the signal doesn't extend too far.
Second, make sure you disable broadcast of the SSID. Although this does little to prevent hackers from connecting to the network, it's still considered best practice for securing a wireless network.
Third, you can enable MAC Filtering on the WAP. With MAC Filtering, the administrator must go in and list the MAC addresses of all the devices that are allowed to connect to the wireless network. Unfortunately, it's easy for an attacker to discover the MAC addresses of wireless clients. It's also easy to program a network card to use a specific MAC address rather than the one assigned by the manufacturer. When you program a NIC to use a different MAC address, this is called MAC Spoofing. MAC Filtering isn't great wireless security, but it can be helpful. It's also the only way to address problems with rogue (unauthorized) devices connecting to the wireless network.
To properly protect a wireless network, you must use wireless encryption.

802.11 Encryption Standards

Different versions of 802.11 introduced features for securing wireless networks.
The first encryption offered for wireless networks was the Wired Equivalent Privacy (WEP) which used the RC4 encryption algorithm. Unfortunately, the way WEP implemented RC4 was flawed, and it was easily hacked.
In 2003 the IEEE released an update for WEP called Wi-Fi Protected Access (WPA). WPA also used RC4, but it implemented Temporal Key Integrity Protocol (TKIP). WPA with TKIP improved on the encryption in WEP. Because it used the same encryption algorithm, WEP devices could easily be upgraded to WPA. However, WPA was intended only as a temporary fix.
In 2004, with 802.11i, IEEE released WPA2. WPA2 uses the Advanced Encryption Standard (AES). Instead of TKIP, WPA2 uses Counter Mode Cipher Block Chaining Message Authentication Code Protocol (CCMP) which provides better encryption.
In 2018, WPA3 was released. WPA3 uses even better encryption. It also provides better mechanisms for ensuring messages arrive without being corrupted (integrity). WPA3 also supports perfect forward secrecy (PFS). PFS uses a different key for each session or transaction. That means that if an attacker discovers a key, it can only be used to decrypt a small amount of data.
All the versions of WPA support two modes. Personal mode uses a Pre-Shared Key (PSK). The user inputs a code on the WAP. To connect to the WAP, the same key must be input on the client. Most SOHO networks use the Personal mode.
Enterprise mode supports 802.11x which allows port authentication. When setting a WAP to Enterprise mode, you must provide the IP address of a Remote Authentication Dial In User Service (RADIUS) server. When the client connects to the WAP, it launches a captive portal. The user must input a username and password. This information is sent to the WAP which passes it along to the RADIUS server. The RADIUS server contacts an authentication database and tries to authenticate the client. Then it relays the results to the WAP. If the client has been authenticated, the WAP allows it to connect to the wireless network. Enterprise mode is typically only used in business environments.
If your wireless network supports WPA3, that's what you should choose. If not, choose WPA2. WPA and WEP are not recommended.

Wireless Implementation

WAP Placement

All signals are subject to attenuation. Attenuation is the tendency of a signal to degrade over distance. Suppose a child was outside playing. Their parent leans out the window and calls them inside for dinner. If the child is playing too far from the home, they won't hear their parent calling. A child playing nearby will hear with no problems.

The same is true for wireless signals. Depending on the 802.11 standard being used, and the power settings on the WAP, at some distance the wireless won't be in range. The only solution is to get close to the WAP.

Before placing the WAP, you should do a site survey. A site survey is just an inspection of the site to note sources of interference like walls or fluorescent lights. Other WAPs can also interfere. You can use a Wi-Fi Analyzer to create a heat map. A heat map is a graphic representation of signal strength. Then, you can place the WAP in the best location. Performing a site survey can also help you decide how many WAPs you might need to provide good wireless coverage.

Wi-Fi Antennas

Most wireless equipment uses omni-directional antennas. If you've ever seen a wireless access point or NIC that has an antenna that looks like a stick, that's an omni-directional antenna. These antenna's send the signal out in all directions.

If you need to supply wireless coverage in a narrower area or concentrated in a particular direction, you can get a directional antenna. These antennas can be yagi antenna's, which look like the old television antennas. They have multiple elements of different lengths. The signal is concentrated in the direction of the shortest element. They also make mini-parabolic dishes (they look like satellite dishes) for wireless.

Yagi Antenna

Extending a Wireless Network

There are multiple options if you need to extend wireless coverage.

It's possible to just implement multiple WAPs to increase coverage. However, each WAP is a different wireless network. That means each WAP will need to have a different SSID. This would require users to manually connect to each WAP as they come in range of the WAP. This will not provide seamless access across a greater distance.

You could configure a wireless extender. Wireless extenders are wireless repeaters. They accept signals from the wireless nodes and repeat them to the main network. When the reply comes from the main wireless network, they repeat the answer back to the nodes. Wireless extenders are relatively inexpensive compared to having multiple WAPs. However, they usually do not have as many configuration options as a WAP. They also use a different SSID than the main wireless network which can be confusing for end users.

If you want to provide seamless wireless coverage over a greater distance than one WAP can cover, your best bet is to implement a wireless mesh network. When you purchase multiple WAPs as part of a mesh network, one WAP functions as the main WAP. The other WAPs function like wireless extenders. However,

the mesh network will use only one SSID. As a user gets closer to one WAP, and further away from the others, the NIC will seamlessly switch to using the closest WAP.

Network Transmissions and hardware Lab

Explore NICs

In this lab, we will explore the speed and duplex of NICs.

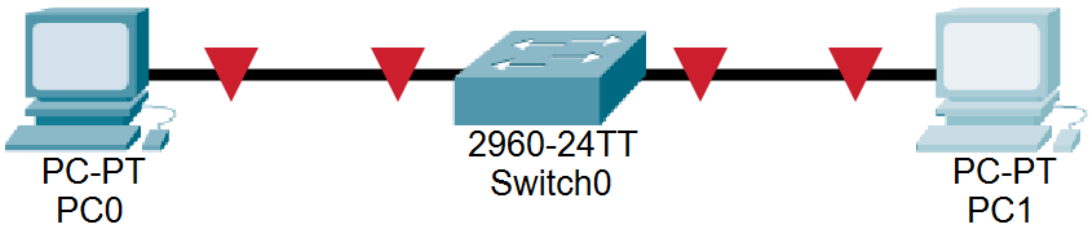
TASK A

1. Download the **3.4.1 Lab File** and open it in **Packet Tracer**.

3.4.1 Lab File

PKT File

2. Notice that the two PCs are not communicating with the switch.



3. Click on **PC0** to open the **PC0 properties** dialog box.

Explore Hubs and Switches

In this lab, we will explore difference in how hubs and switches function.

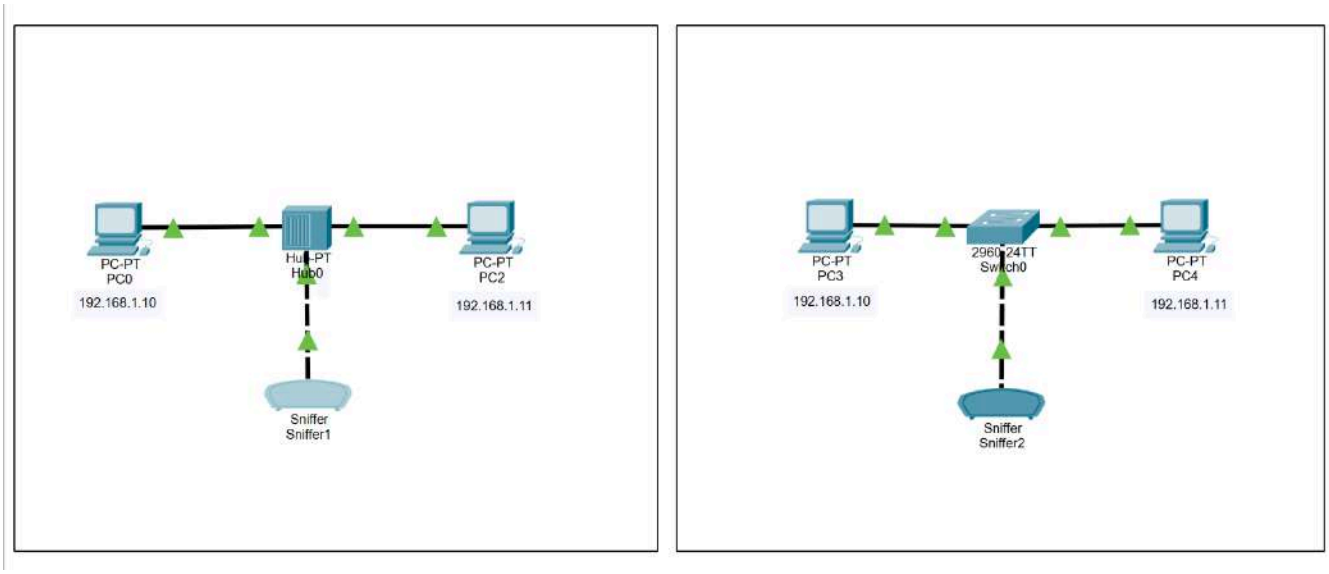
TASK A

In this task, we will look at the function of a hub.

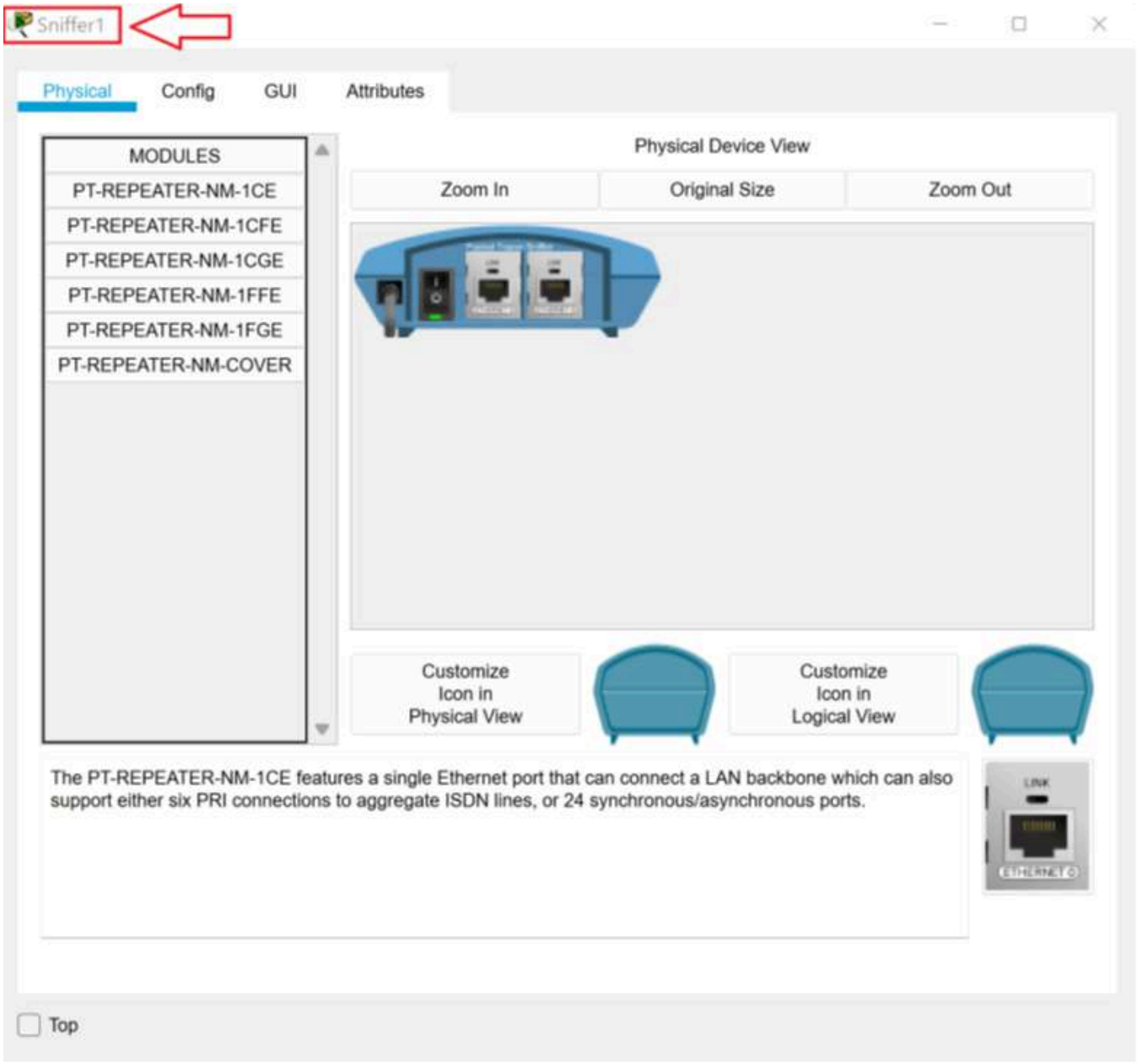
1. Download the **3.4.2 Lab File** and open it in **Packet Tracer**.

3.4.2 Lab File

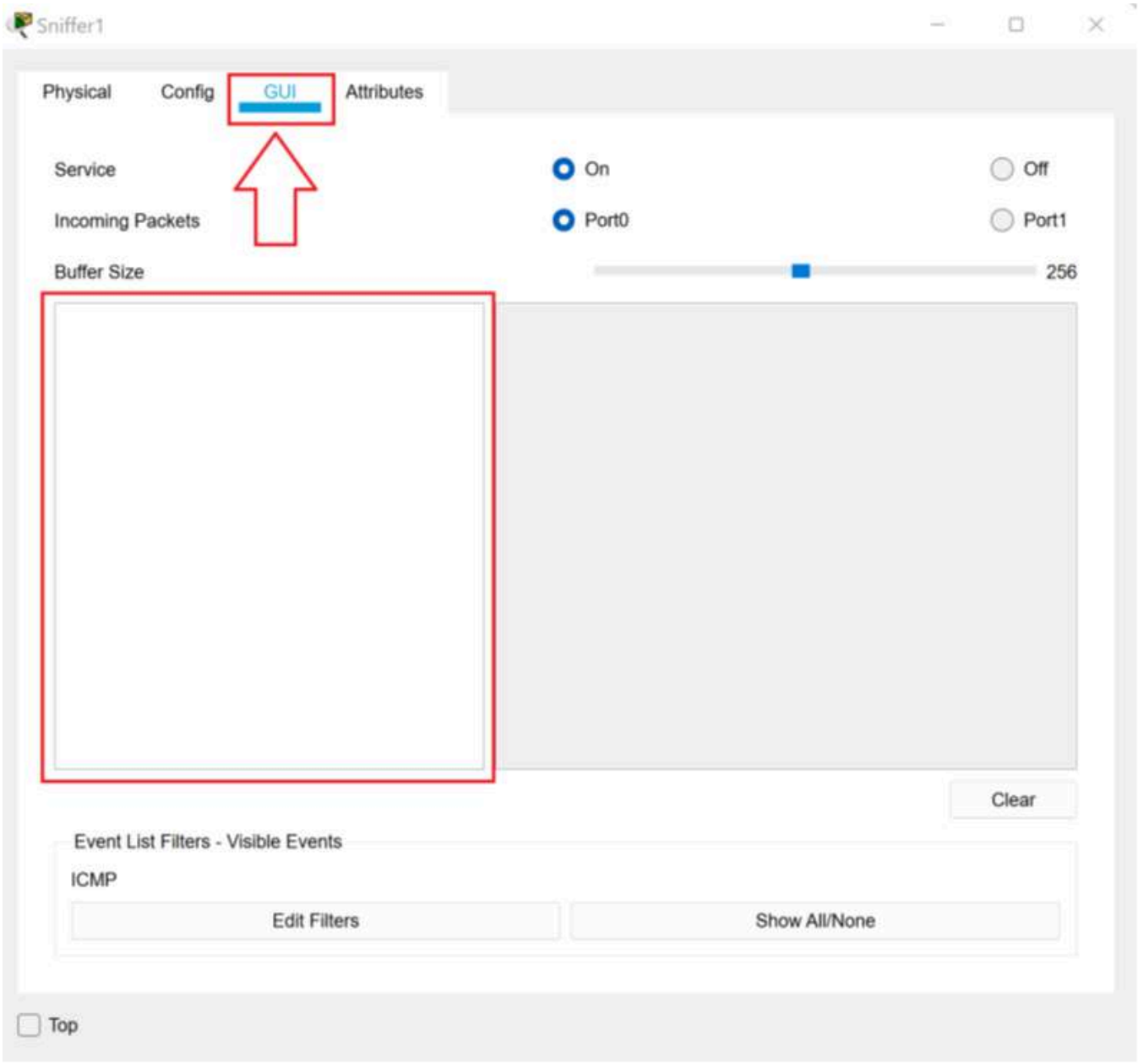
PKT File



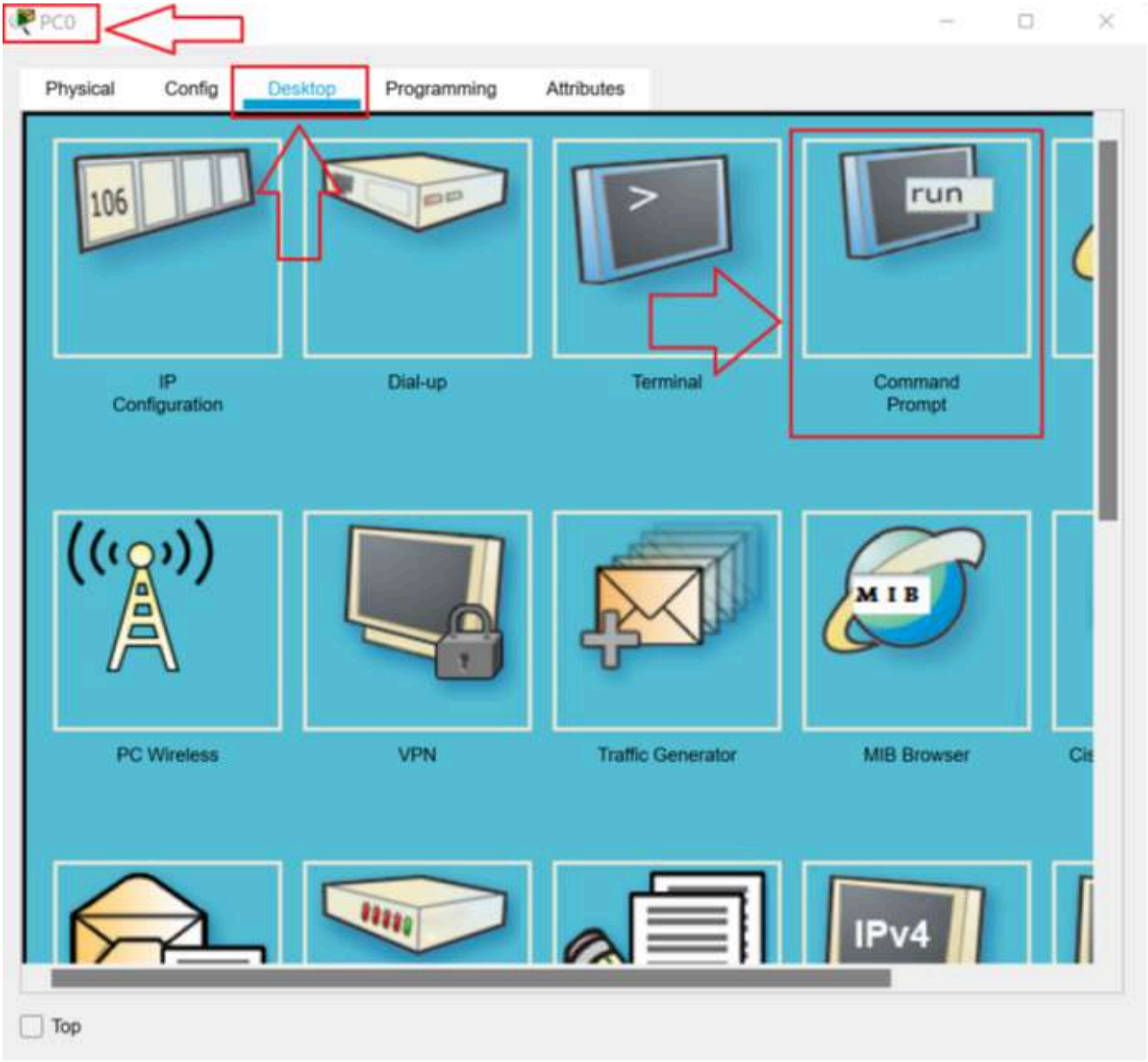
2. Click on **Sniffer1** to open the **Sniffer1 Properties** dialog box.



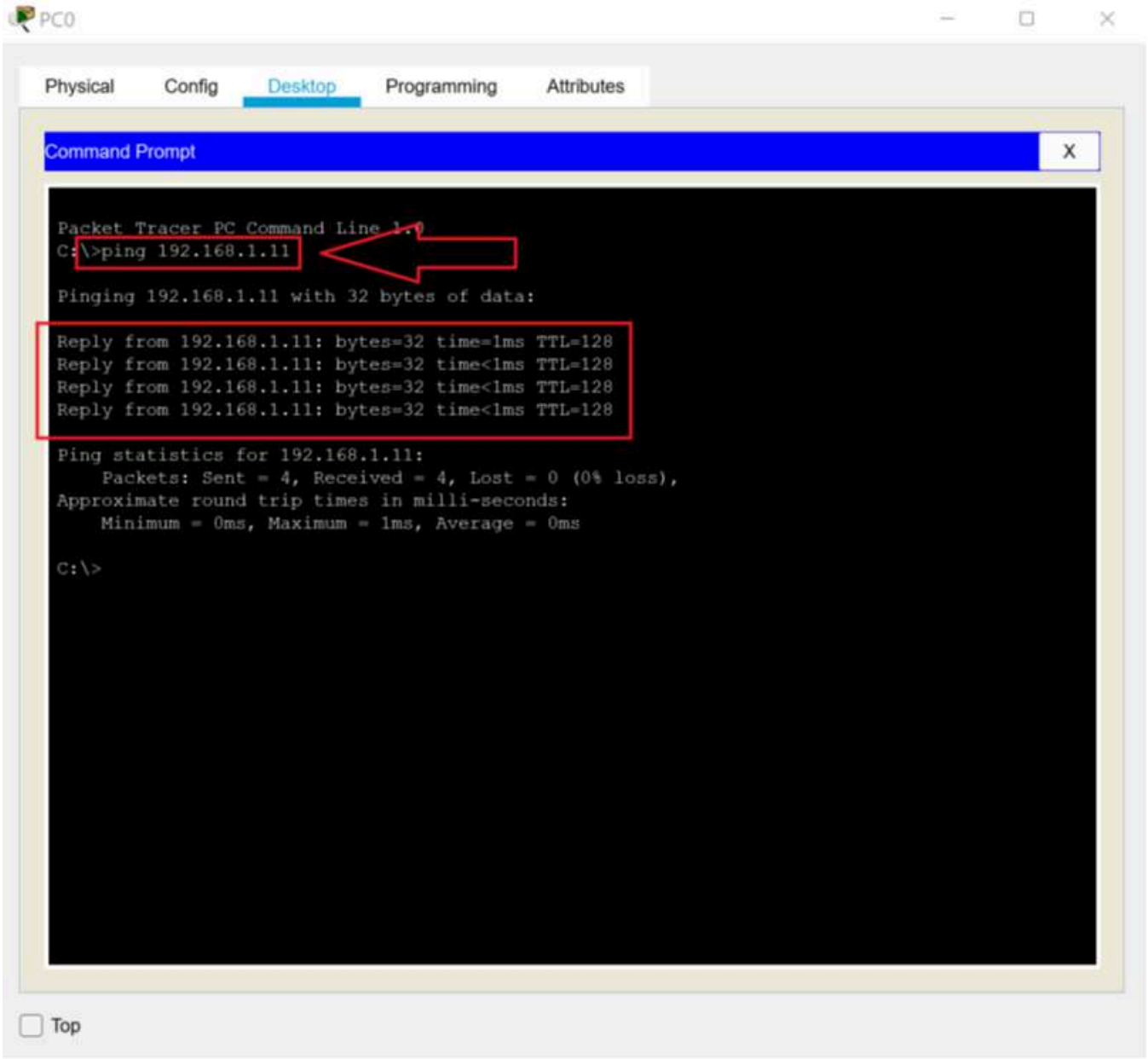
3. Click on the **GUI** tab. (NOTE: If there are any events in the buffer, click the **Clear** button to clear them.)



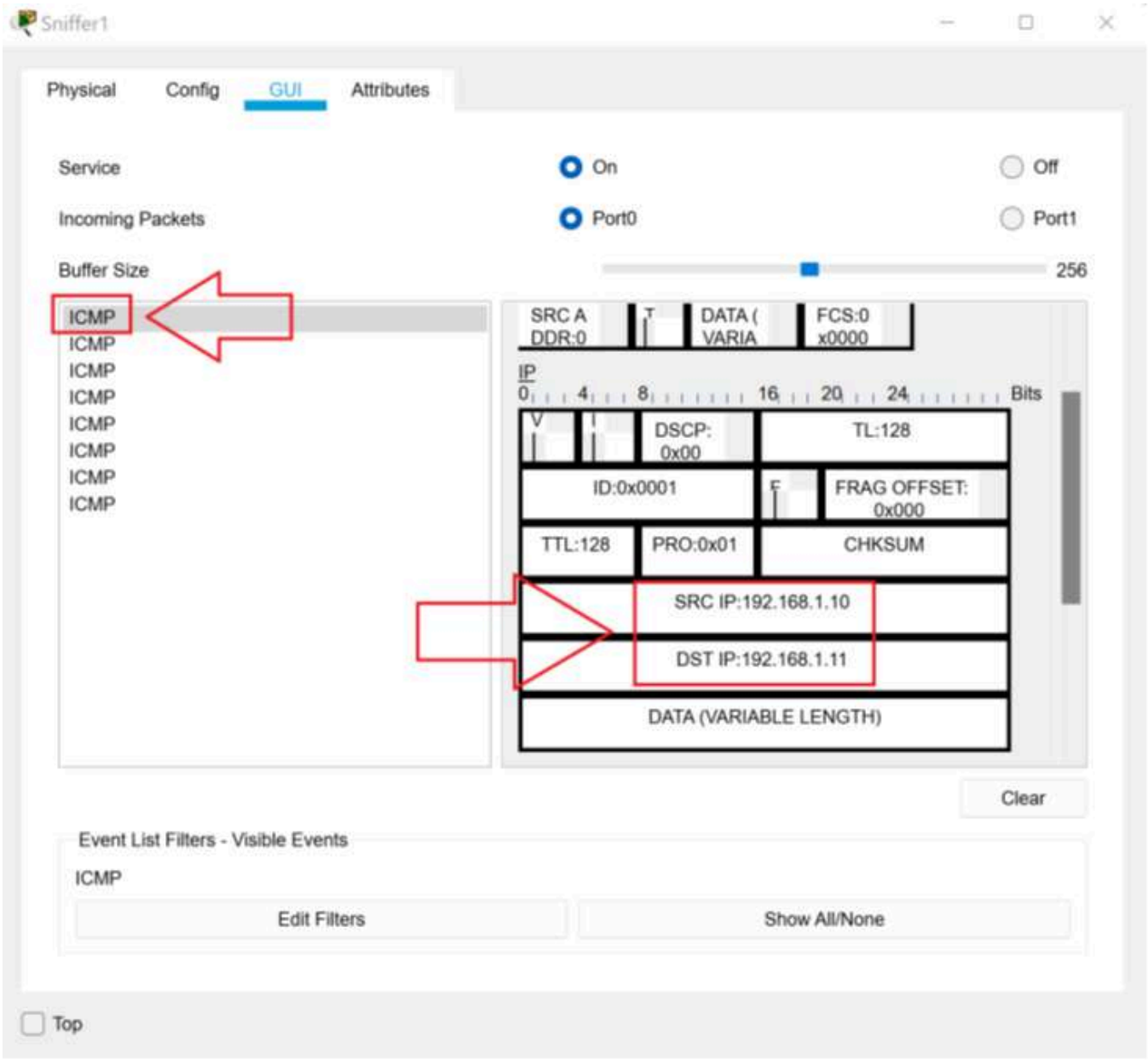
4. Close the **Sniffer1 Properties** dialog box.
5. Click **PC0** to open the **PC0 Properties** dialog box. Then click the **Desktop** tab.
6. Click the **Command Prompt** icon.



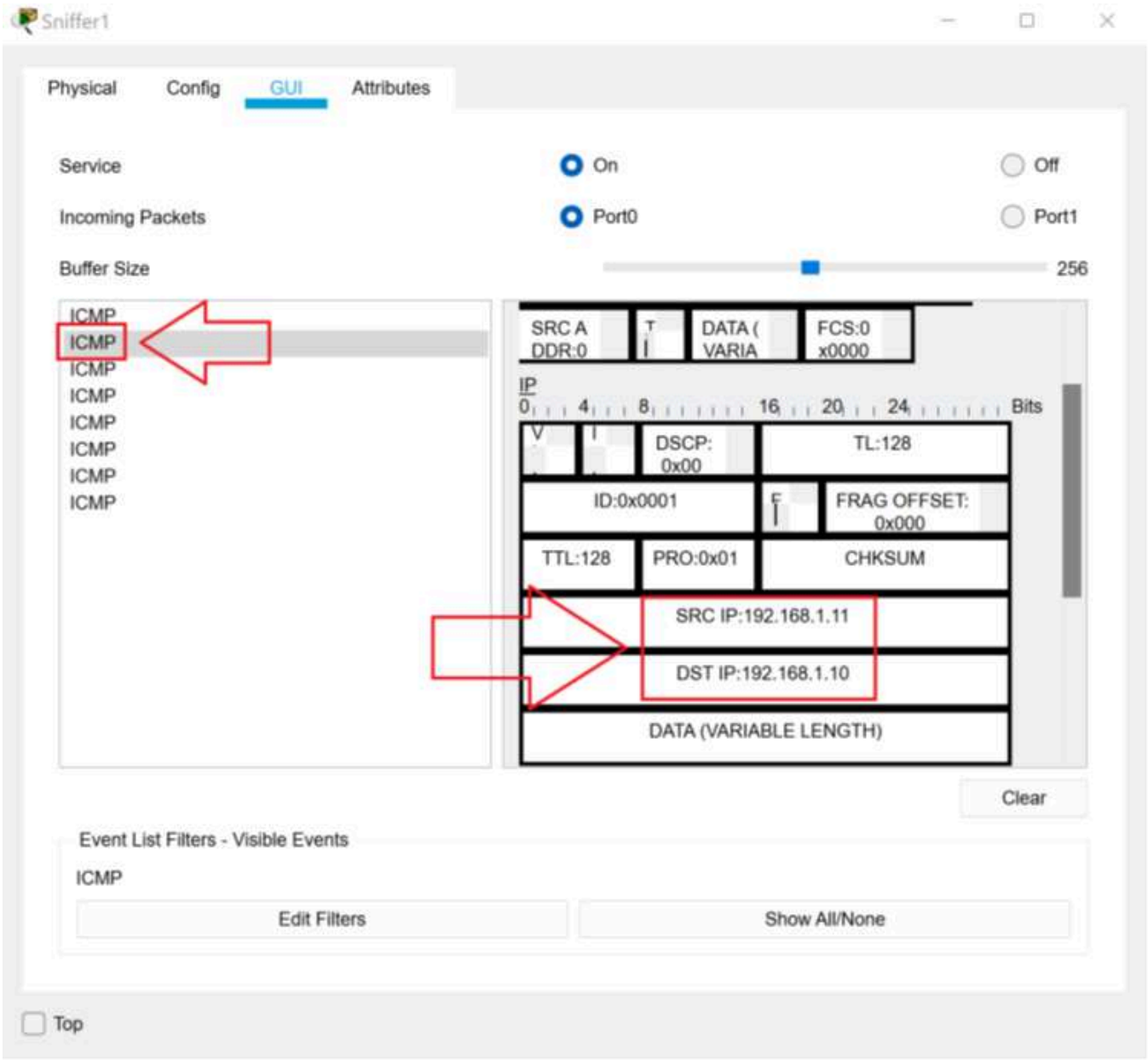
7. In the **Command Prompt**, type **ping 192.168.1.11** and then press **Enter**. You should get four replies.



8. Close the **PC0 Properties** dialog box.
9. Click on **Sniffer1** to open the **Sniffer1 Properties** dialog box. You should see eight ICMP packets. If you click on the first packet, you can see it is the first **ICMP** Echo request from 192.168.1.10 to 192.168.1.11.



10. Click on the second packet. Notice it is the first ICMP Echo Reply from 192.168.1.11 to 192.168.1.10.

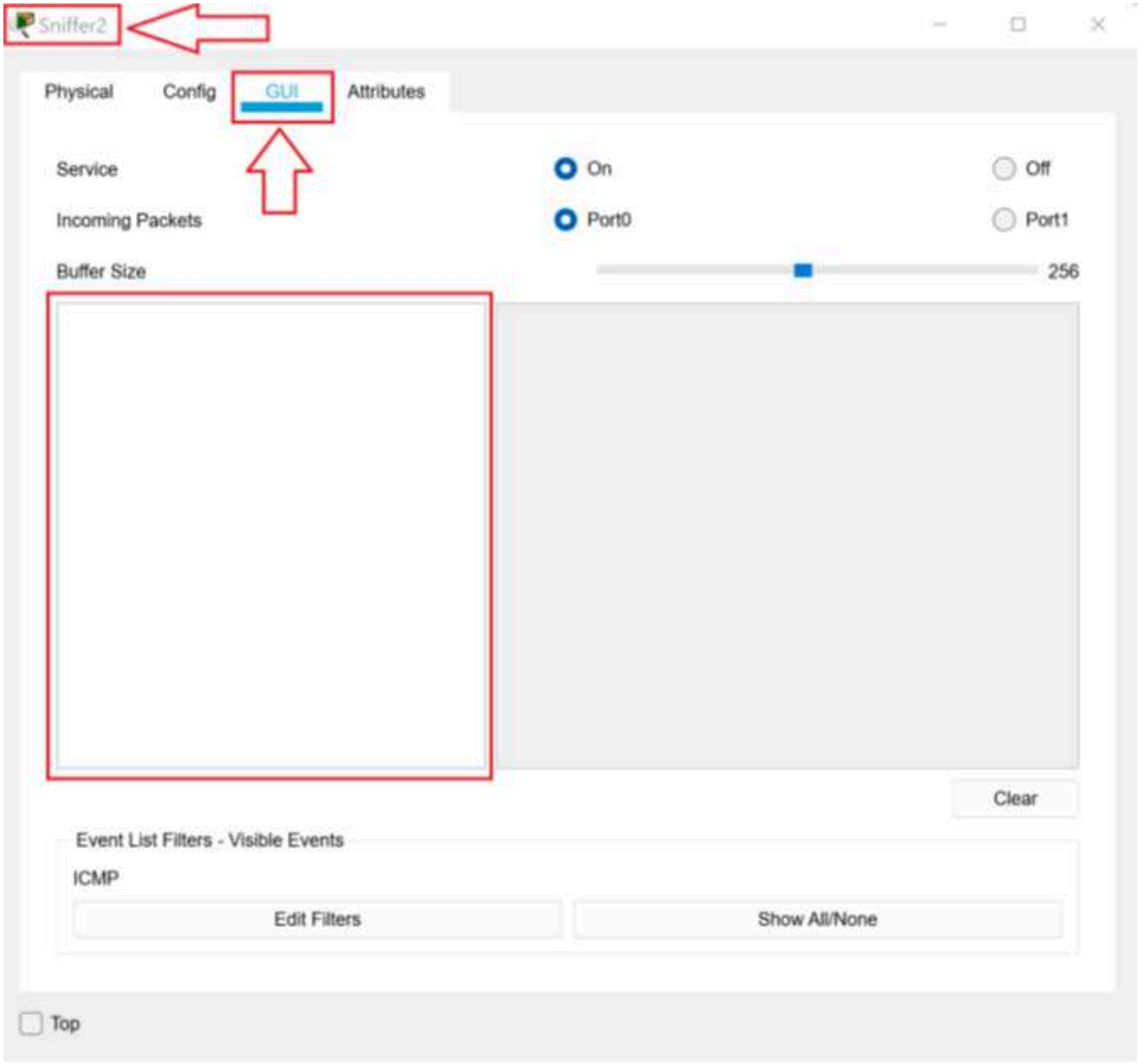


11. The packet sniffer captured the four requests from 192.168.1.10 and the four replies from 192.168.1.11 because the hub sent all of the packets to all of the ports.

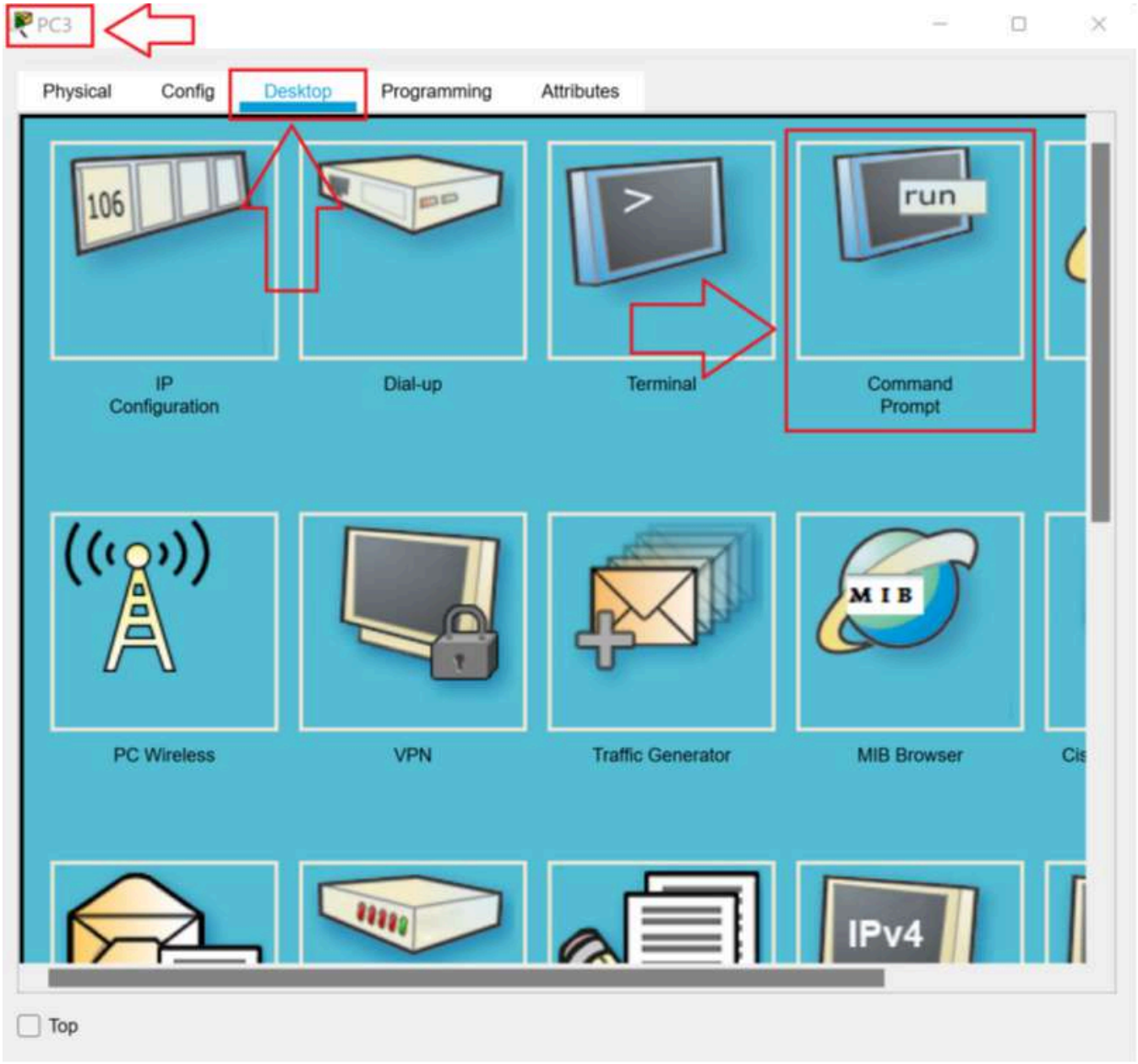
TASK B

In this task, we will look at the function of a switch.

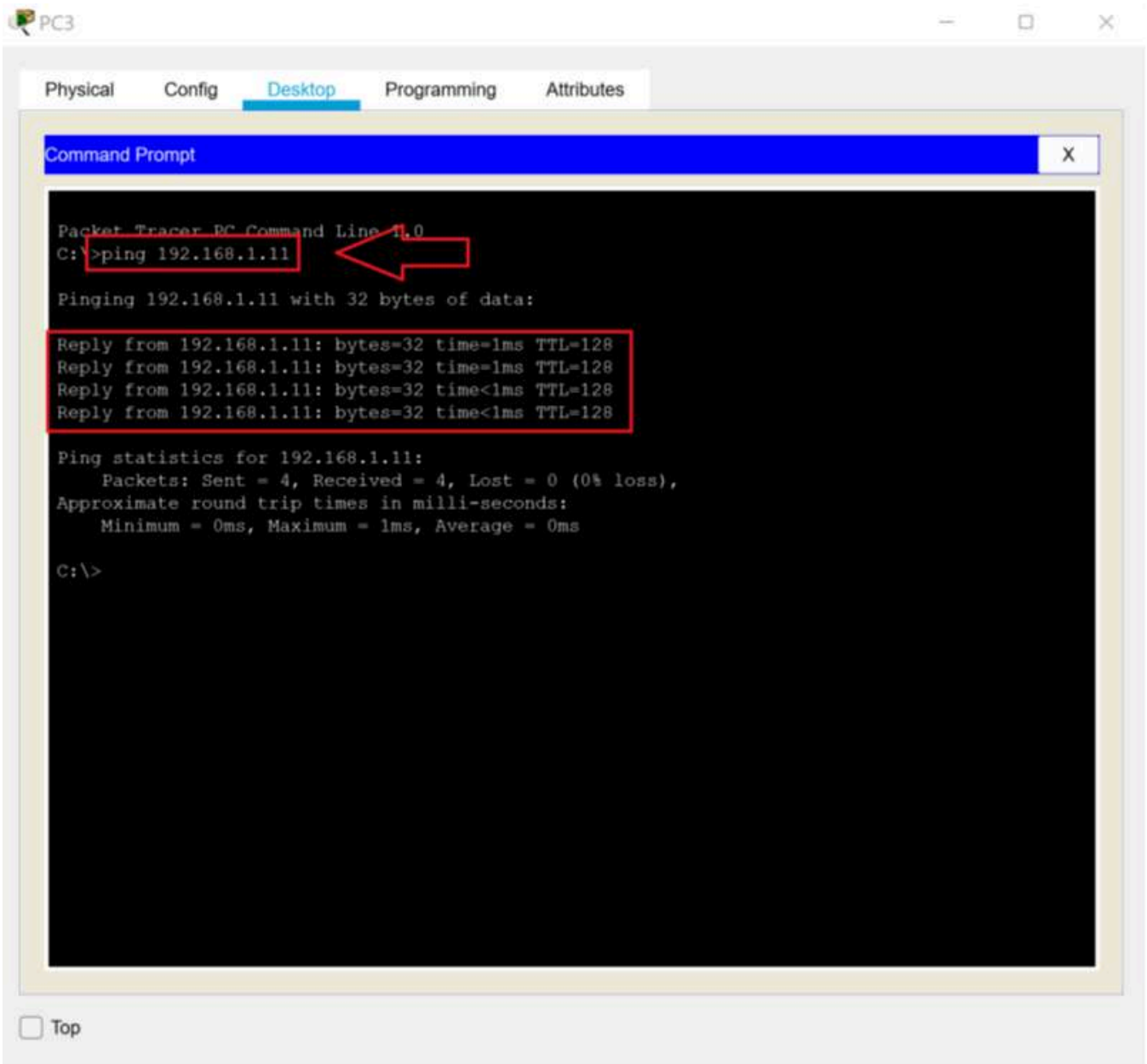
1. Click on Sniffer2 to open the Sniffer2 Properties dialog box. Click on the **GUI** tab. (NOTE: If there are any events in the buffer, click the **Clear** button to clear them.)



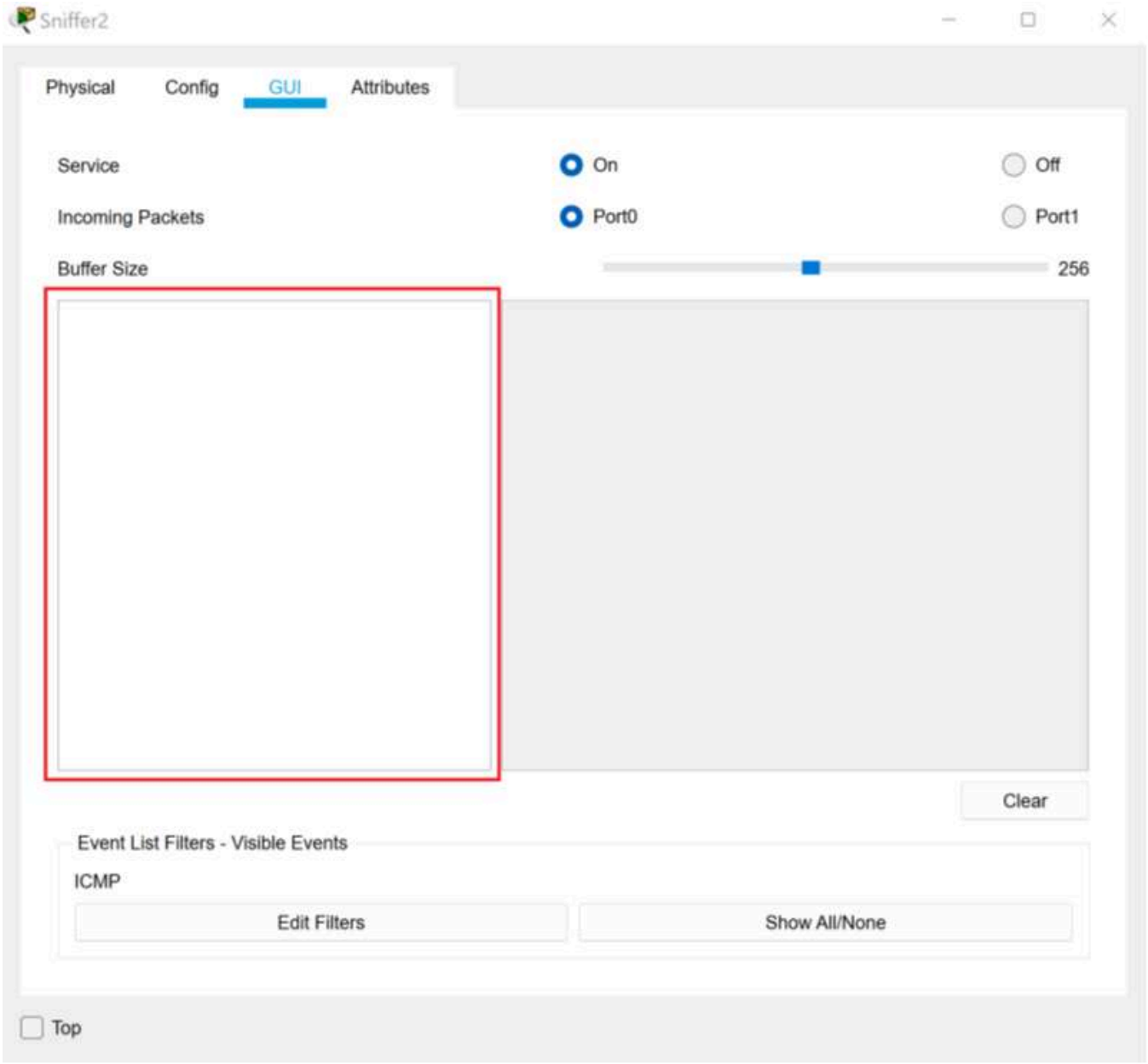
2. Close the **Sniffer2 Properties** dialog box.
3. Click on **PC3** to open the **PC3 Properties** dialog box. Click the **Desktop** tab and then click the **Command Prompt** icon.



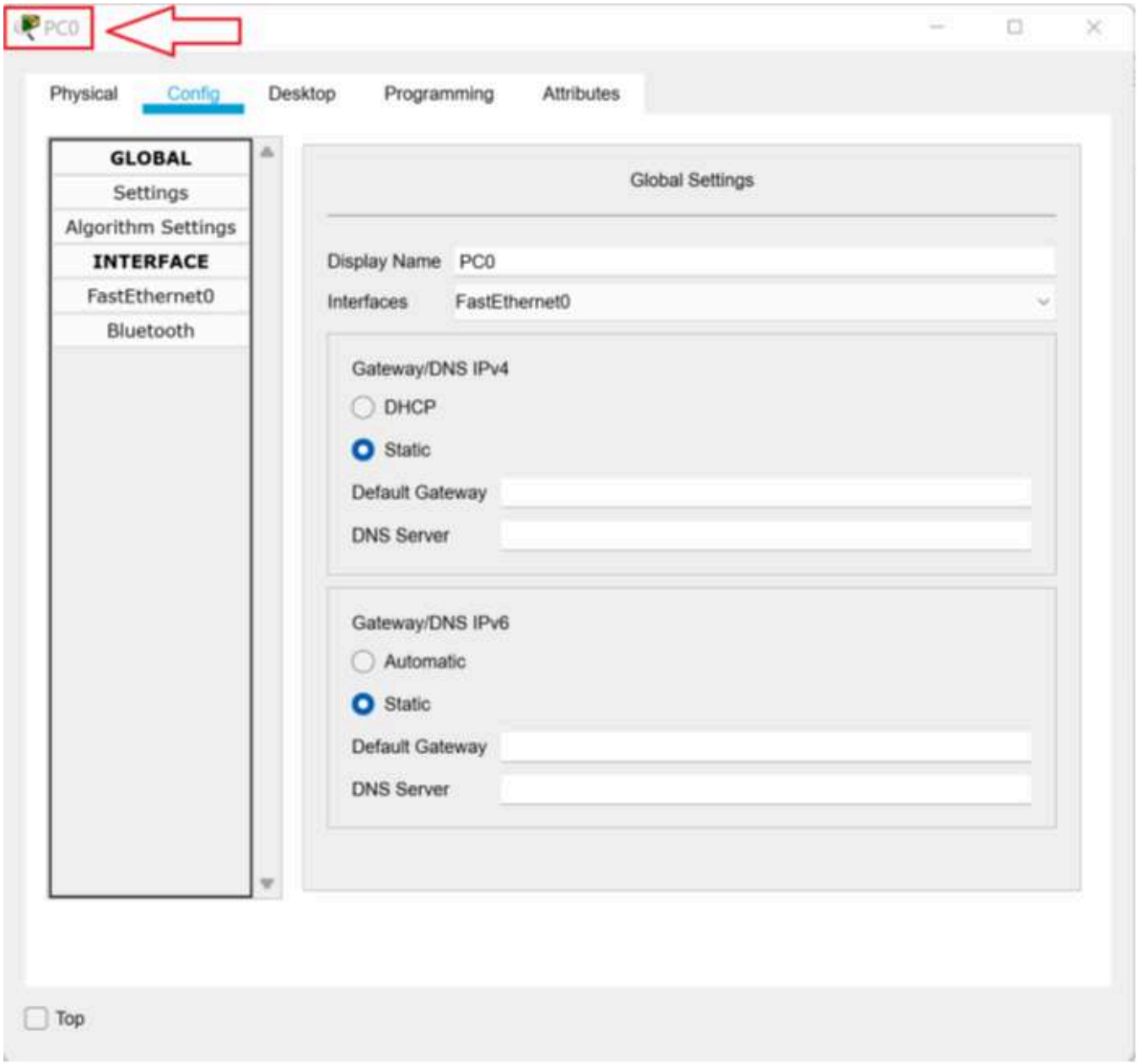
4. In the **Command Prompt**, type **ping 192.168.1.11** and then press **Enter**. You should get four replies.



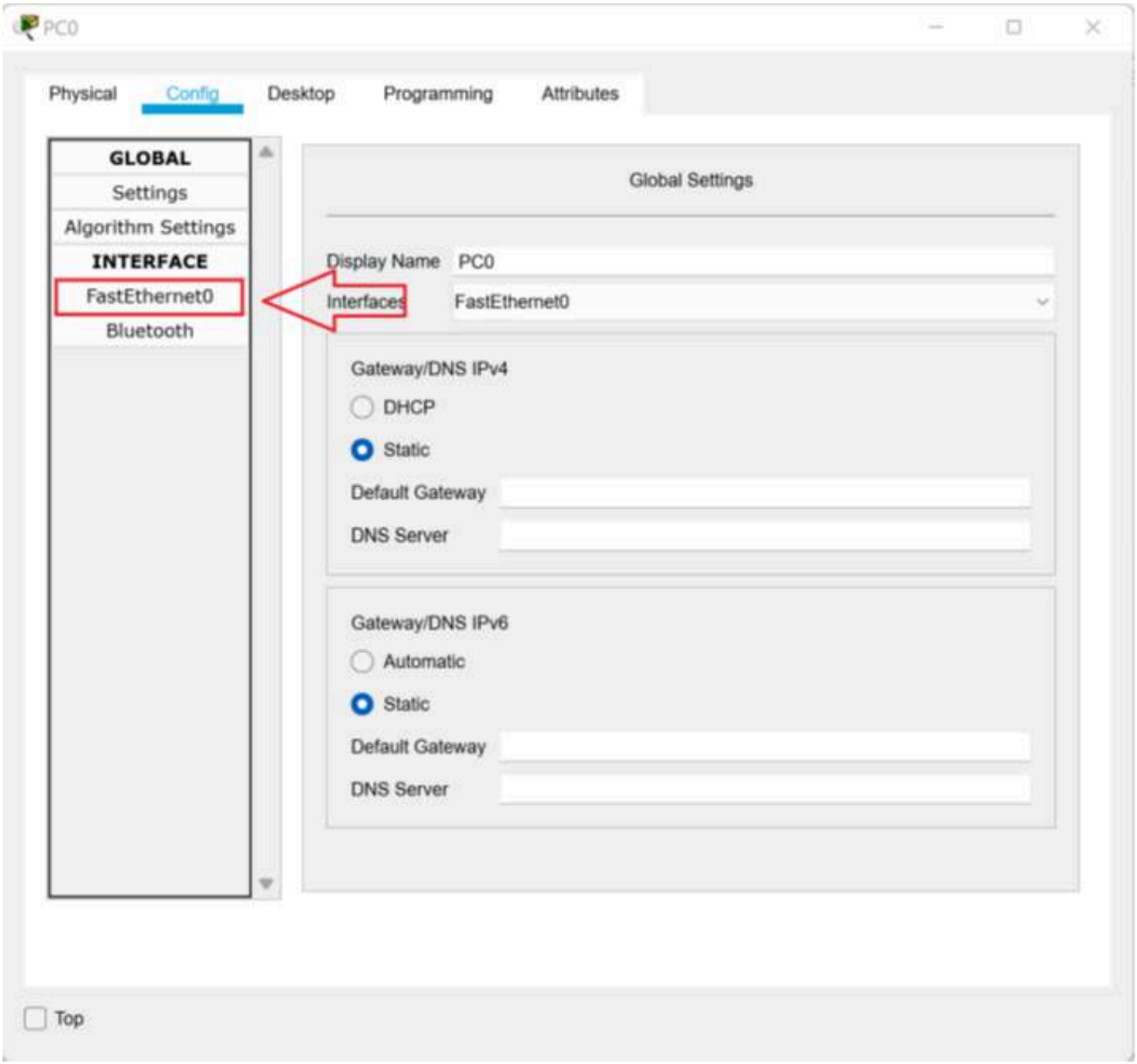
5. Close the **PC3 Properties** dialog box.
6. Click **Sniffer2** to open the **Sniffer2 Properties** dialog box. Notice there are no packets in the buffer. The switch sent the packets directly to the nodes being addressed. Since none of the packets were addressed to the MAC address of Sniffer2, none of the packets were sent to the sniffer.



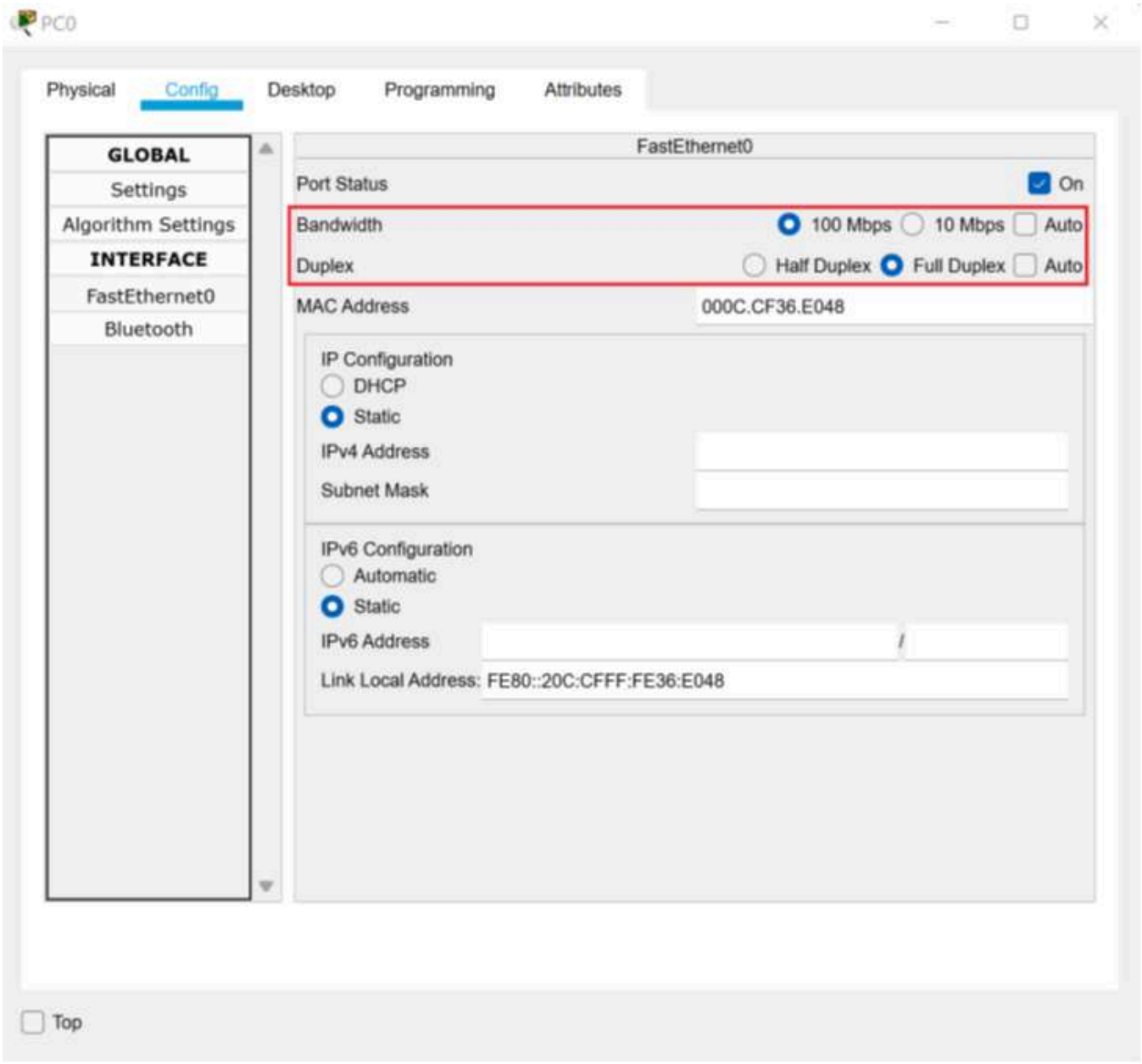
7. Close the **3.4.2 Lab File** file.



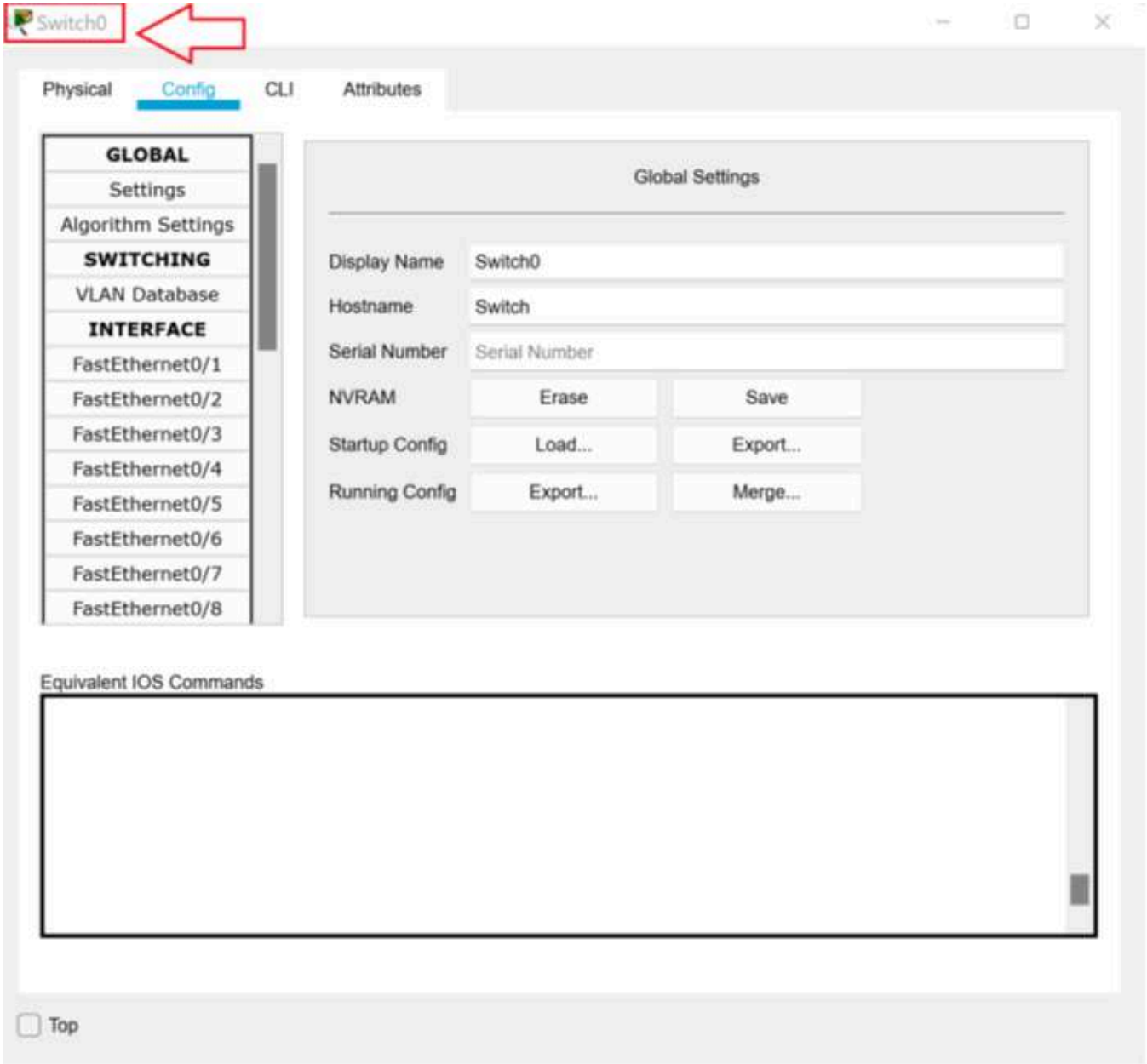
4. In the **Interface** menu, click **FastEthernet0**.



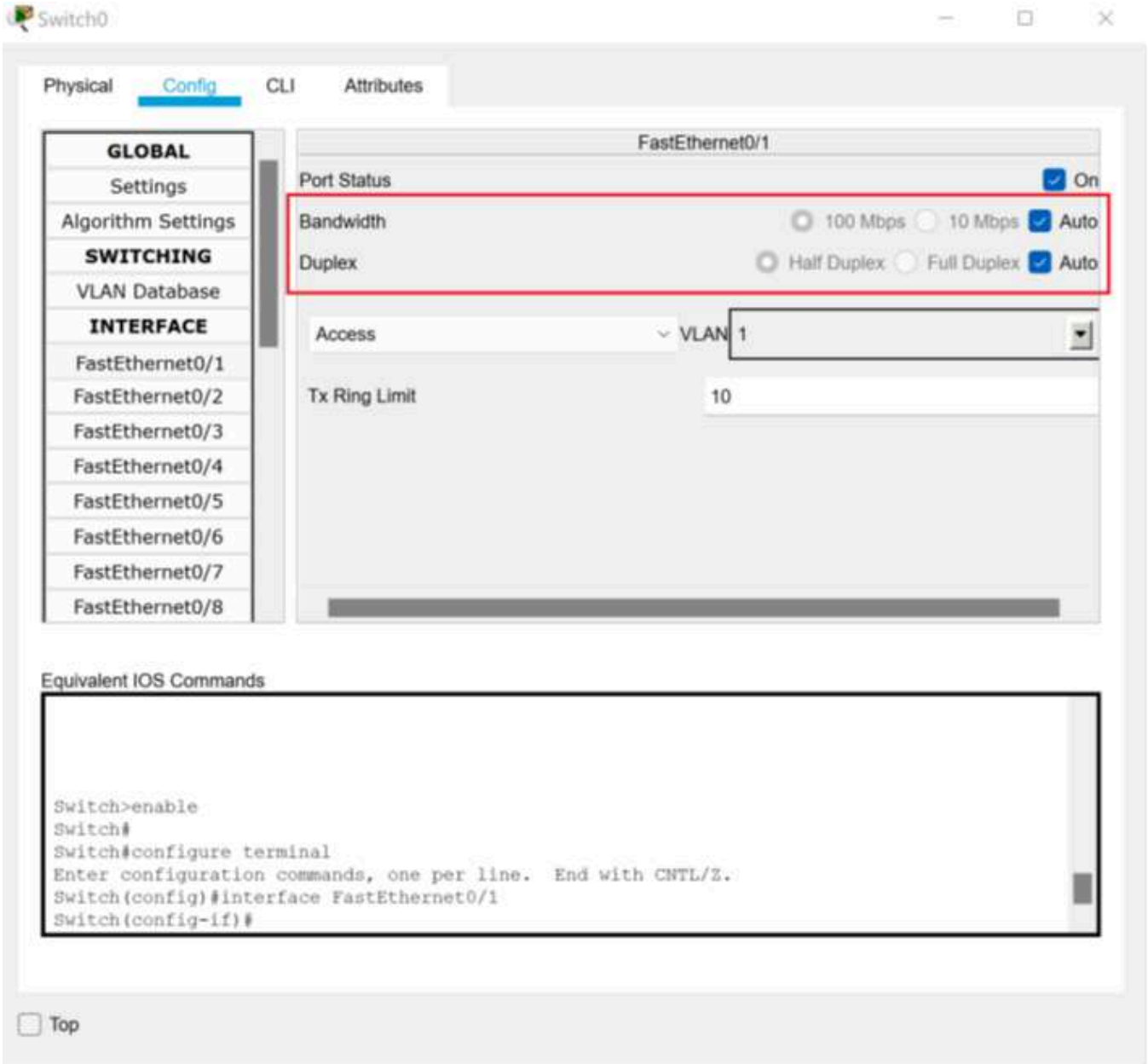
5. Notice the network card is set to **100 Mbps Full Duplex**.



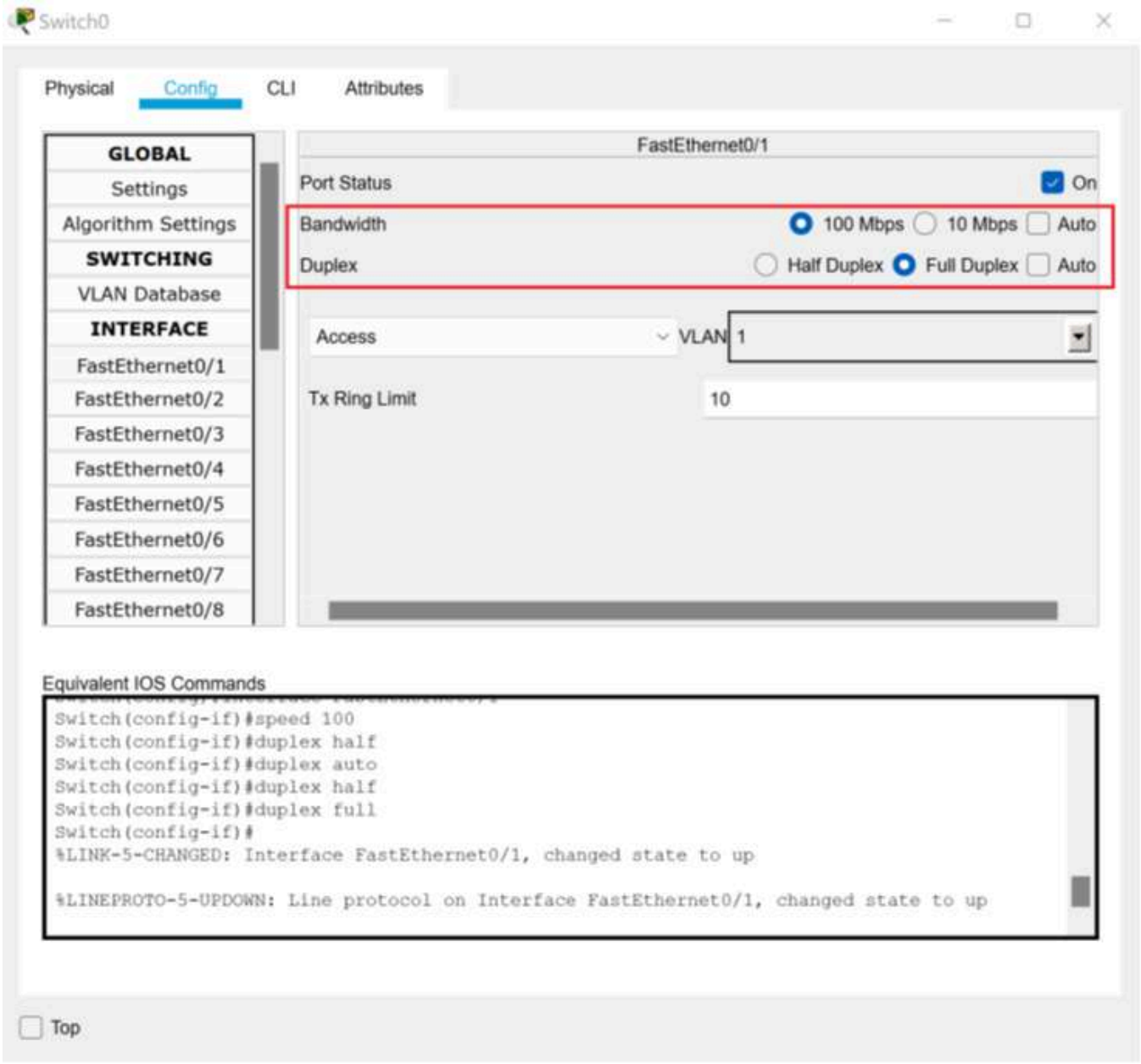
6. Close the **PC0 Properties** dialog box.
7. Click on **Switch0** to open the **Switch0 Properties** dialog box.



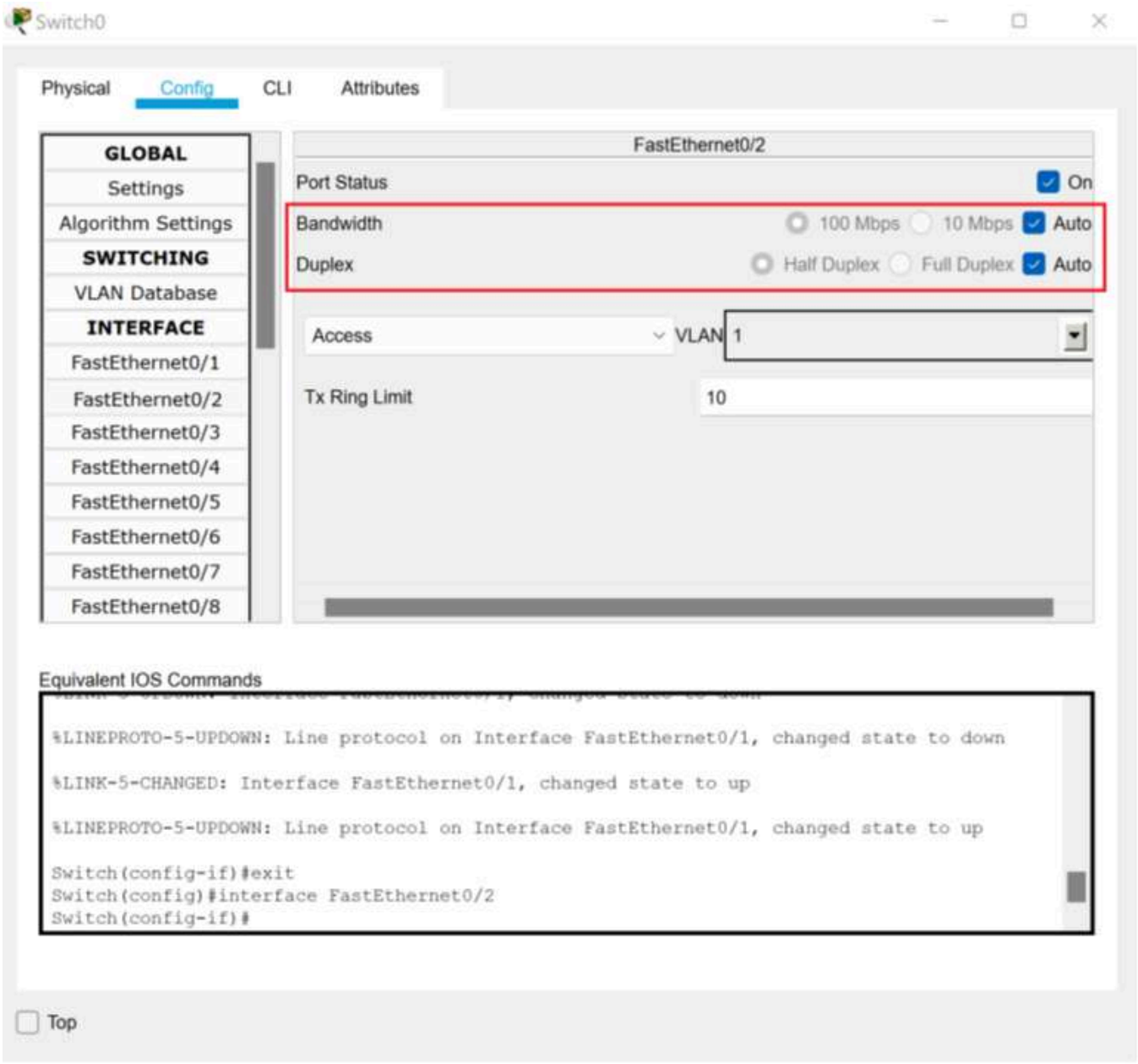
8. In the **Switch0 Properties** dialog box, in the **Interface** menu, click **FastEthernet0/1**. (This is the port on the switch to which the PC0 cable is connected.) Observe that the settings do not match the settings on the client.



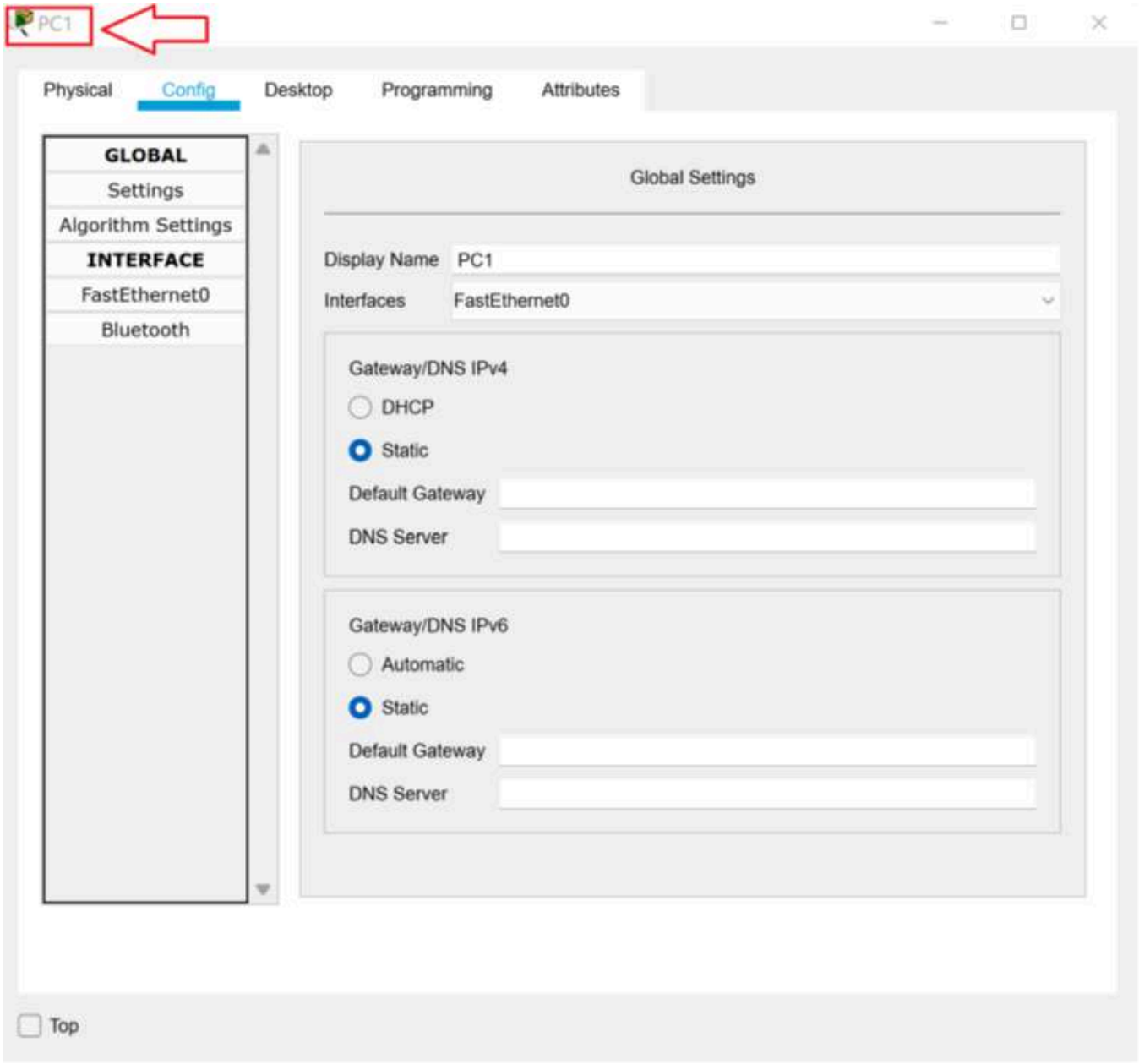
9. To fix the problem, uncheck the **Auto bandwidth** check box. Then uncheck the **Auto Duplex** check box. Select the **Full Duplex** radio button so that the settings on the switch match the settings on the client.



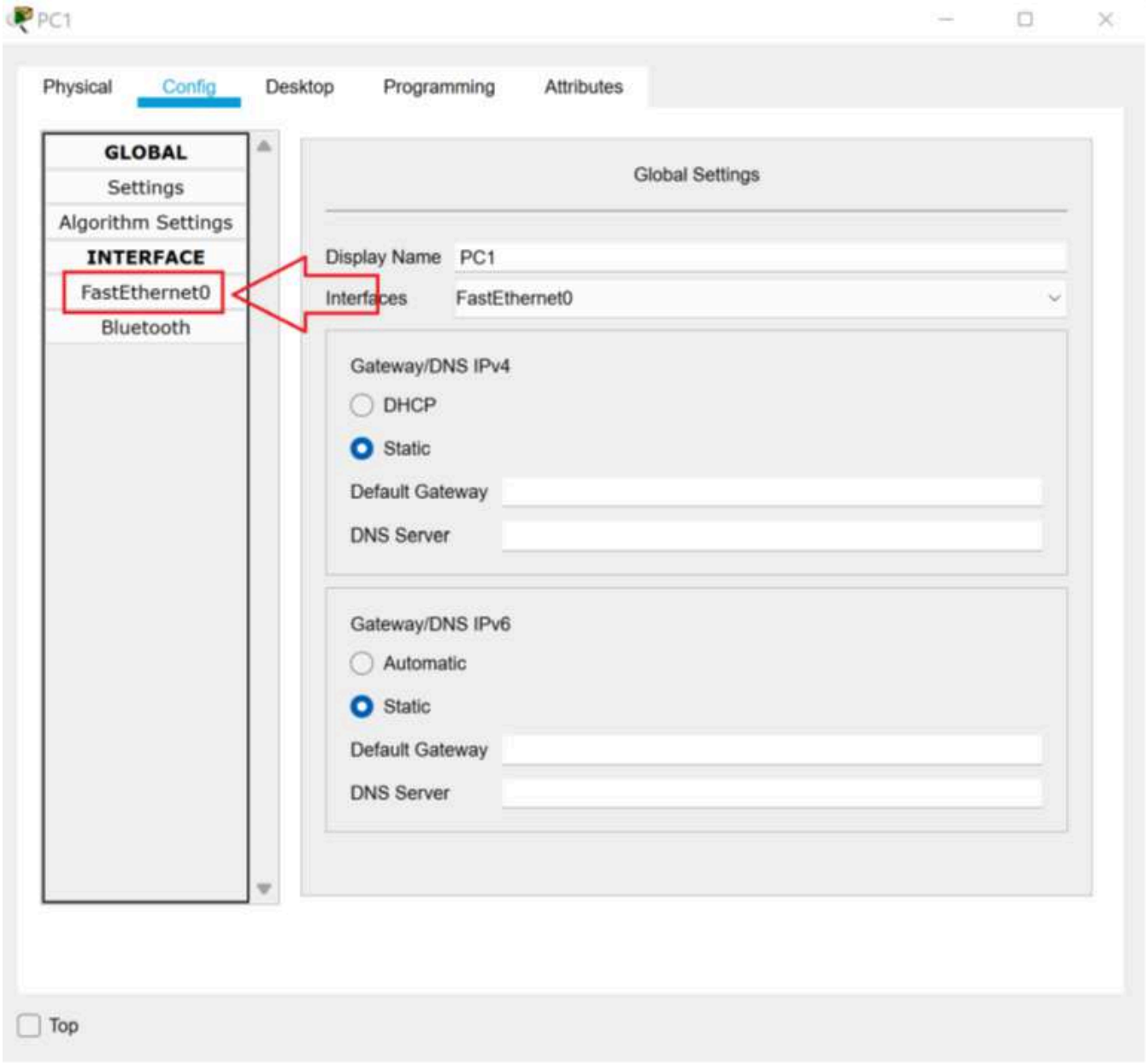
10. Close the **Switch0Properties** dialog box.
11. Notice that the connection between **PC0** and **Switch0** has been resolved.
12. Click **Switch0** to open the **Switch0 Properties** dialog box.
13. In the Interface menu, click **FastEthernet0/2**. (This is the port on the switch to which the PC1 cable is connected.) Notice the port is set to **Auto Bandwidth** and **Auto Duplex**.



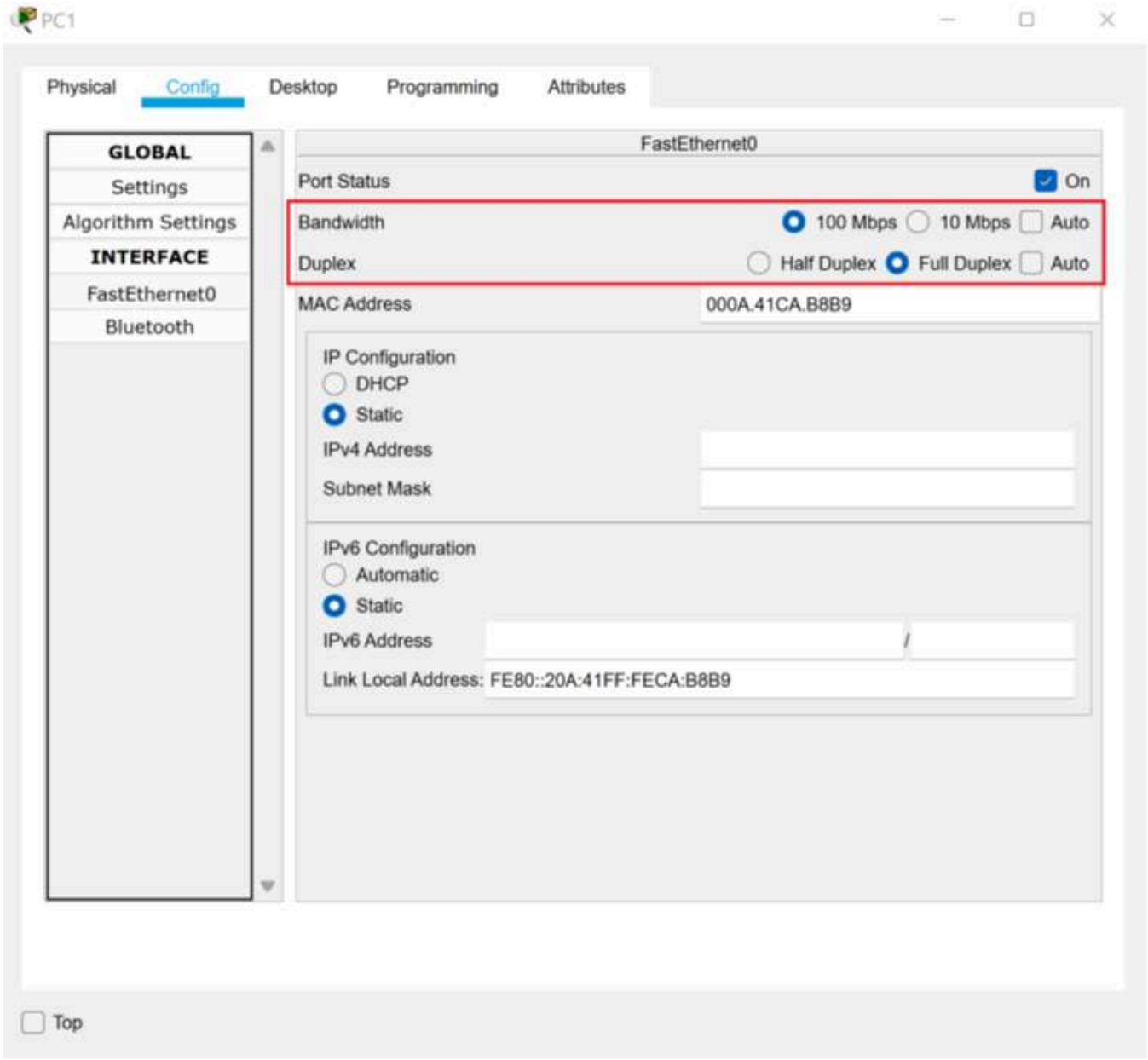
14. Close the **Switch0 Properties** dialog box.
15. Click on **PC1** to open the **PC1 Properties** dialog box.



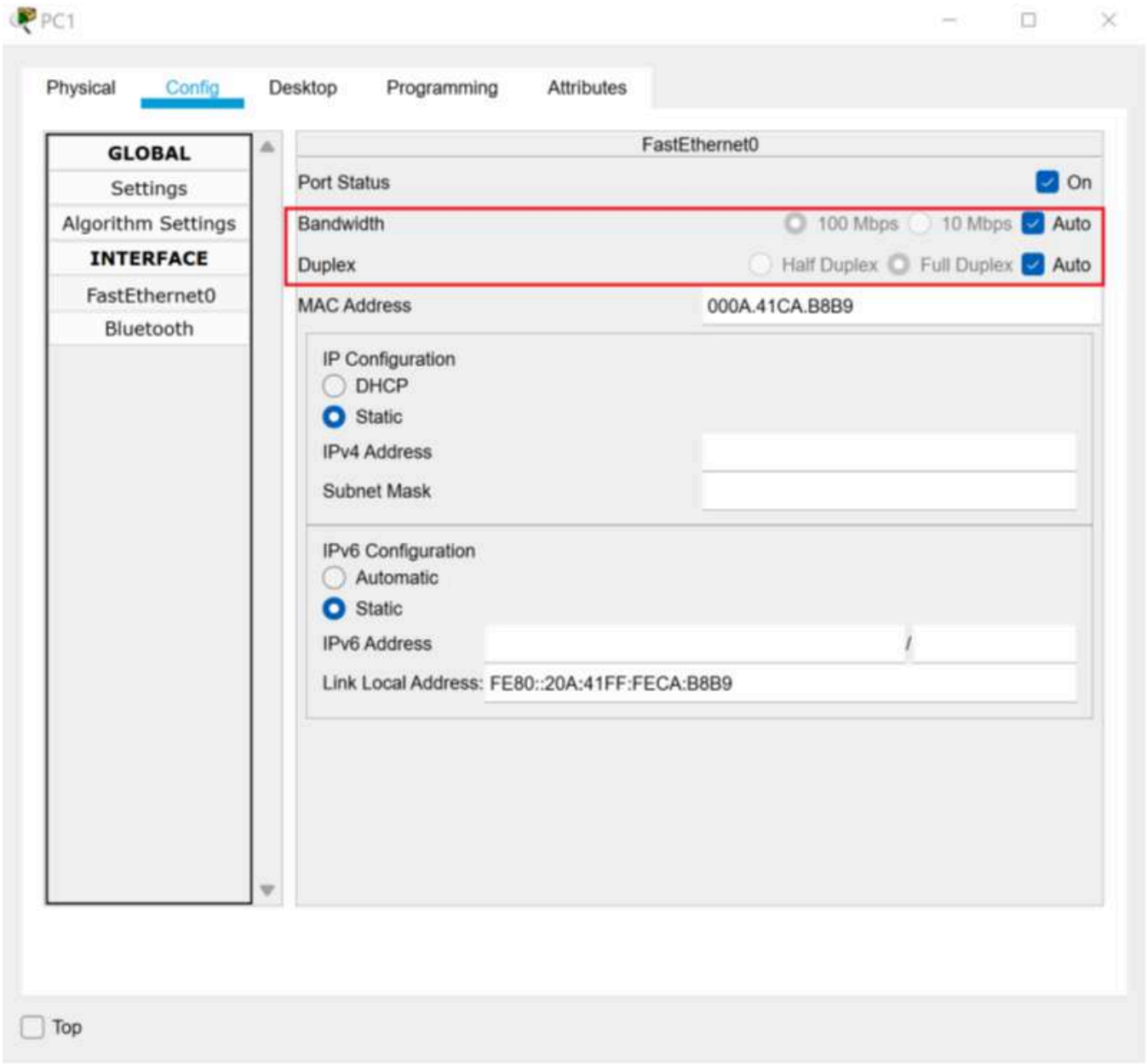
16. In the **Interface** menu, click **FastEthernet0**.



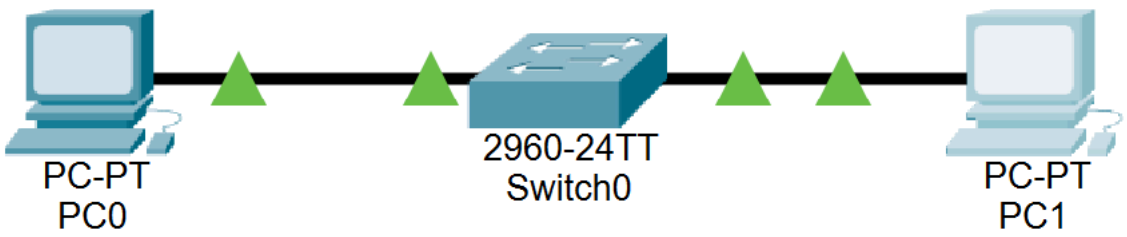
17. Notice the network card is set to **100 Mbps Full Duplex**.



18. To fix the problem, check the **Auto bandwidth** check box. Then check the **Auto Duplex** check box. The settings on the client should now match the settings on the switch port.



19. Close the **PC1 Properties** dialog box.
20. Notice that all problems have been resolved.



21. Close the **3.4.1 Lab File** file.

Explore Routers

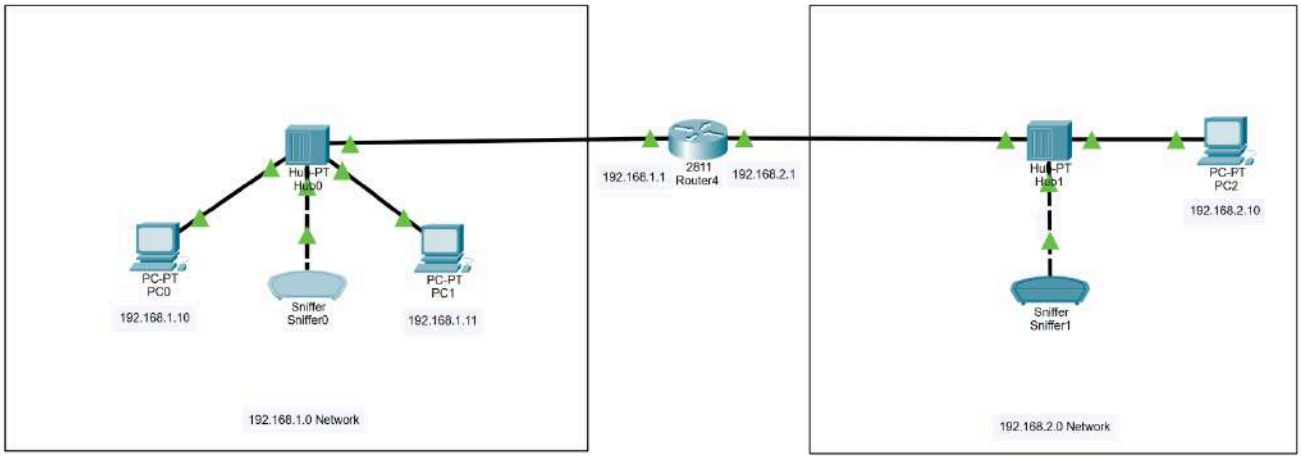
In this lab, we will explore how routers function.

TASK A

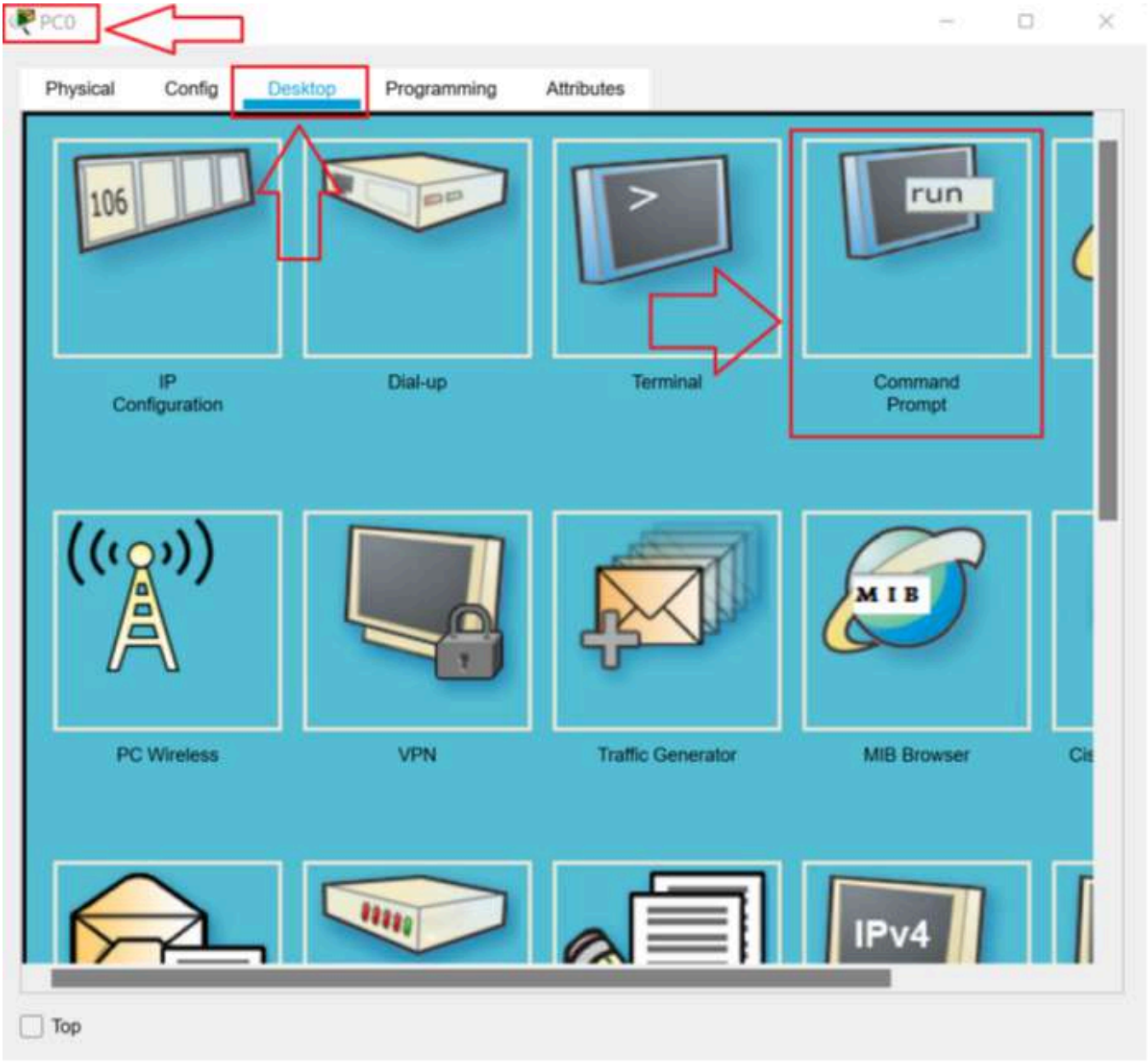
In this task, we will look at how the router handles broadcasts.

1. Download the **3.4.3 Lab File** and open it in **Packet Tracer**.

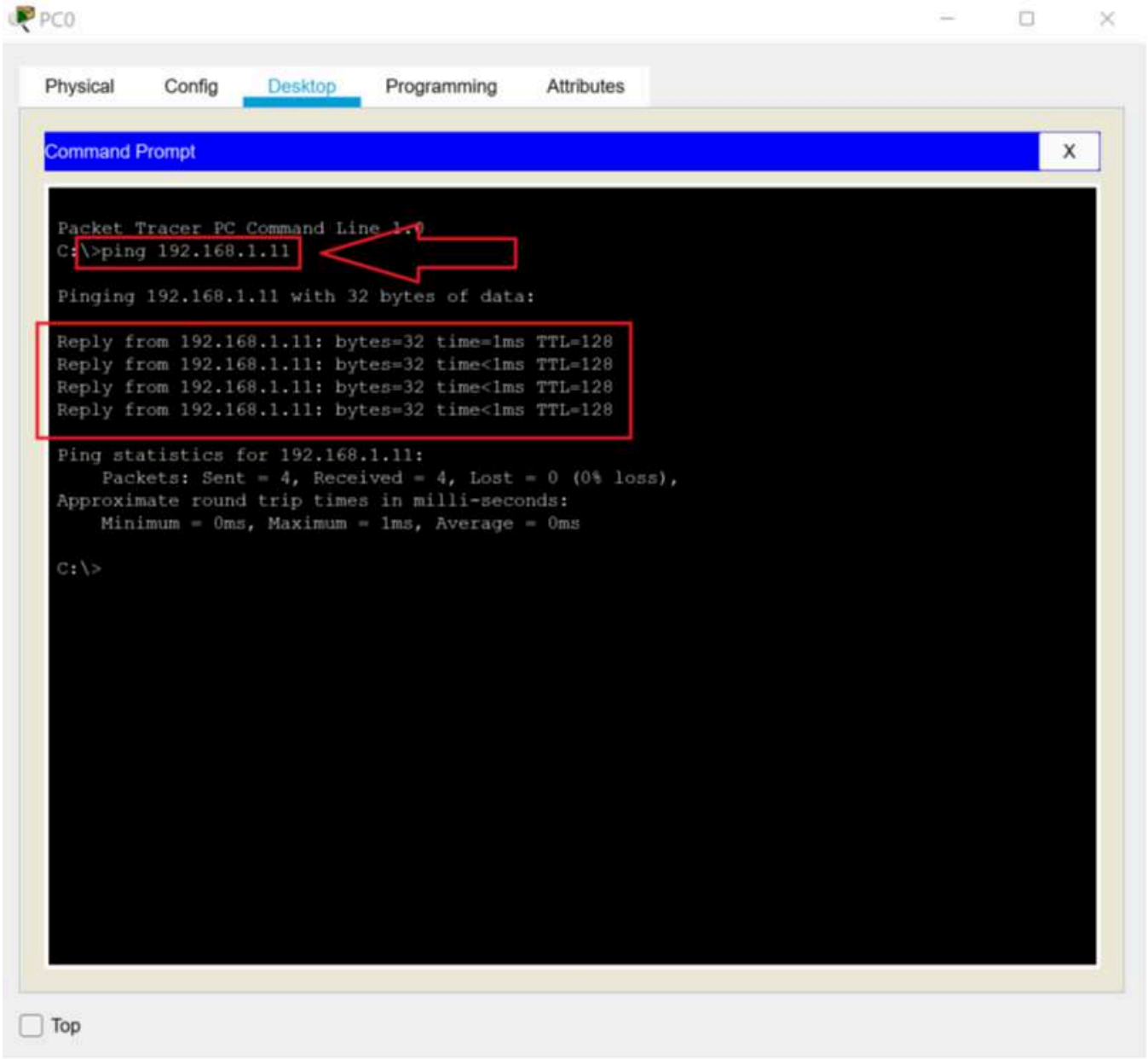
3.4.3 Lab File
PKT File



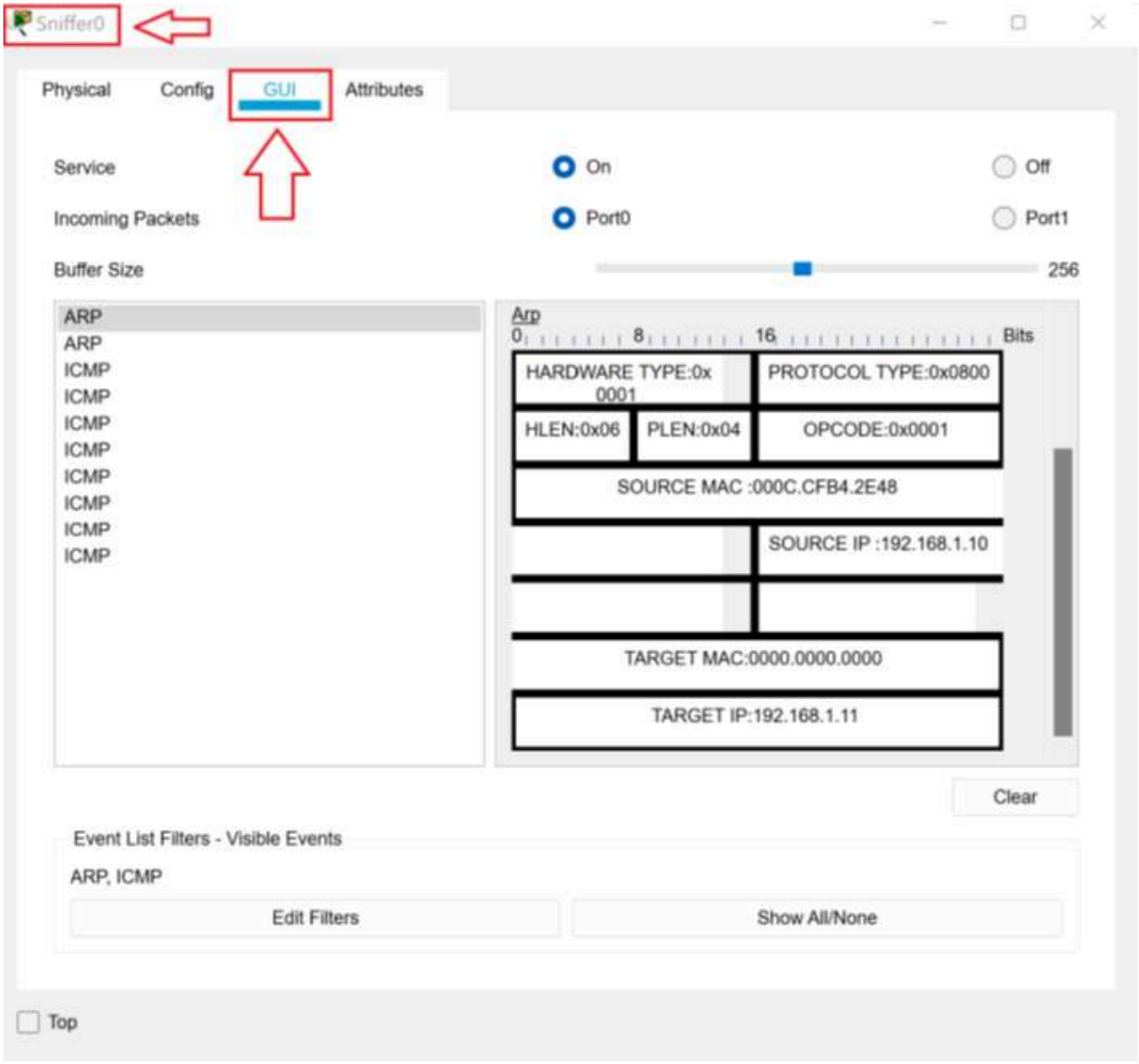
2. Click on **PC0** to open the **PC0 Properties** dialog box. Select the **Desktop** tab, and then click the **Command Prompt** icon.



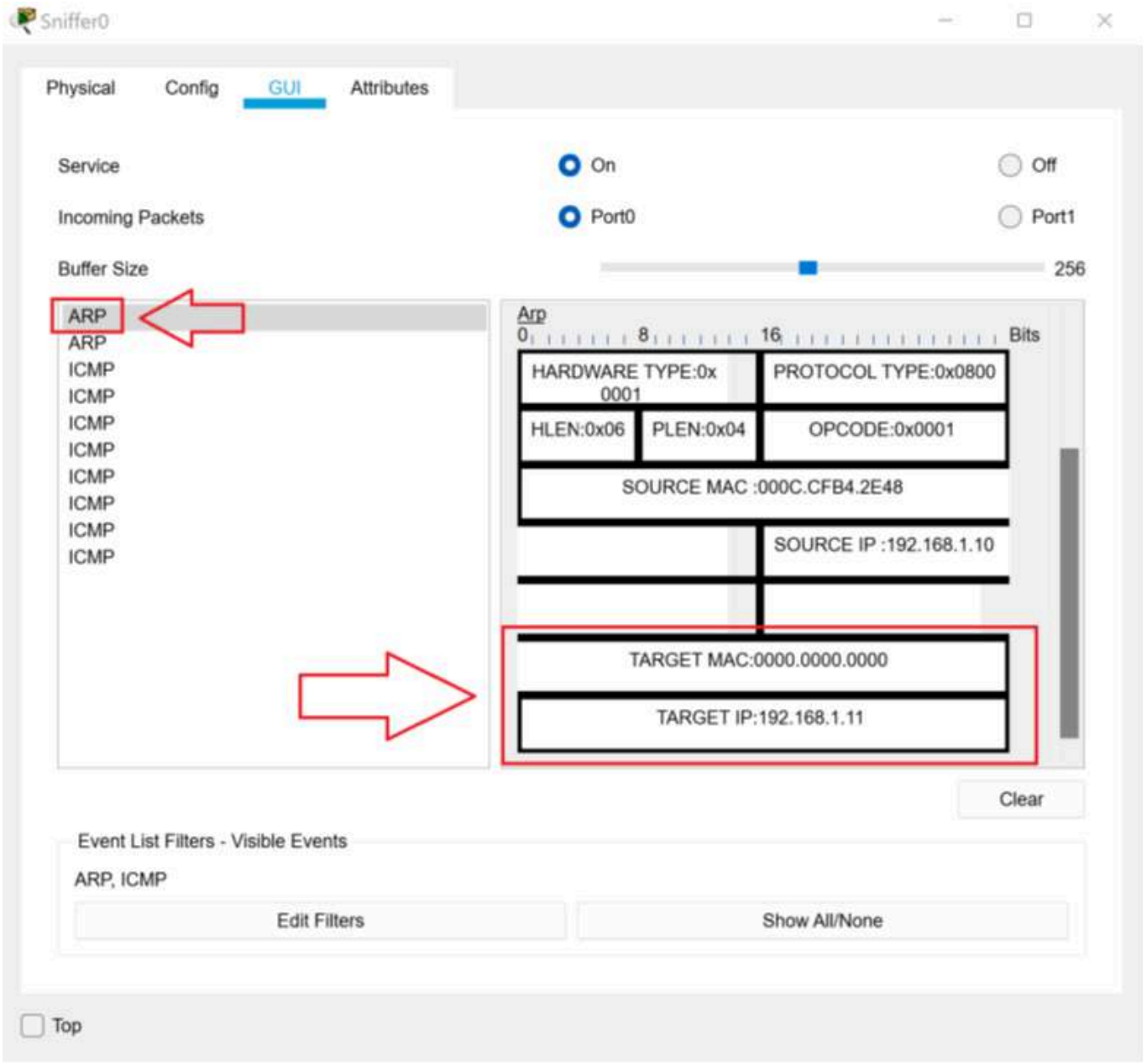
3. In the **Command Prompt**, type **ping 192.168.1.11** and then press **Enter**. You should get four replies.



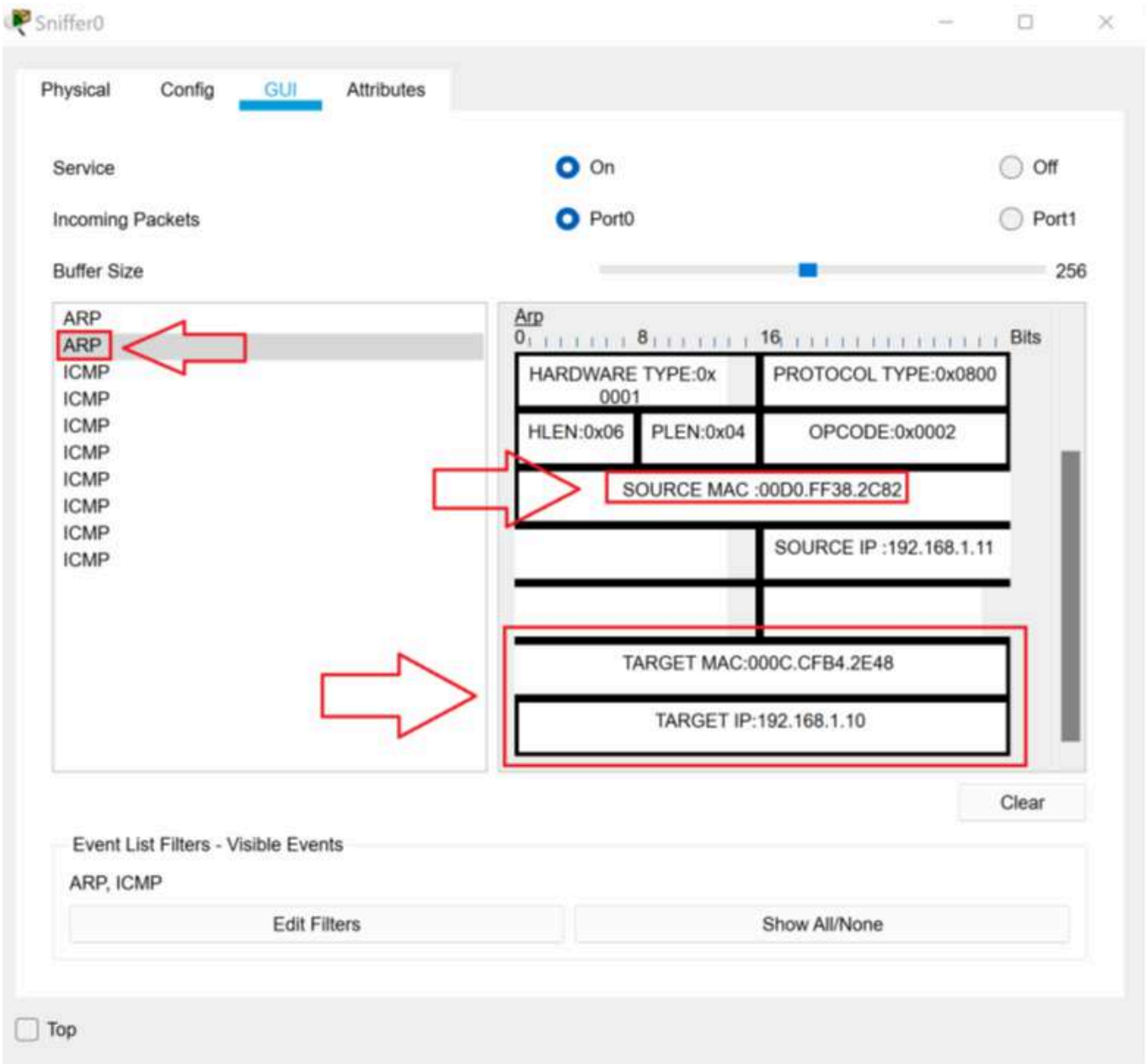
4. Close the **PC0 Properties** dialog box.
5. Click **Sniffer0** to open the **Sniffer0 Properties** dialog box. Then click the **GUI** tab.



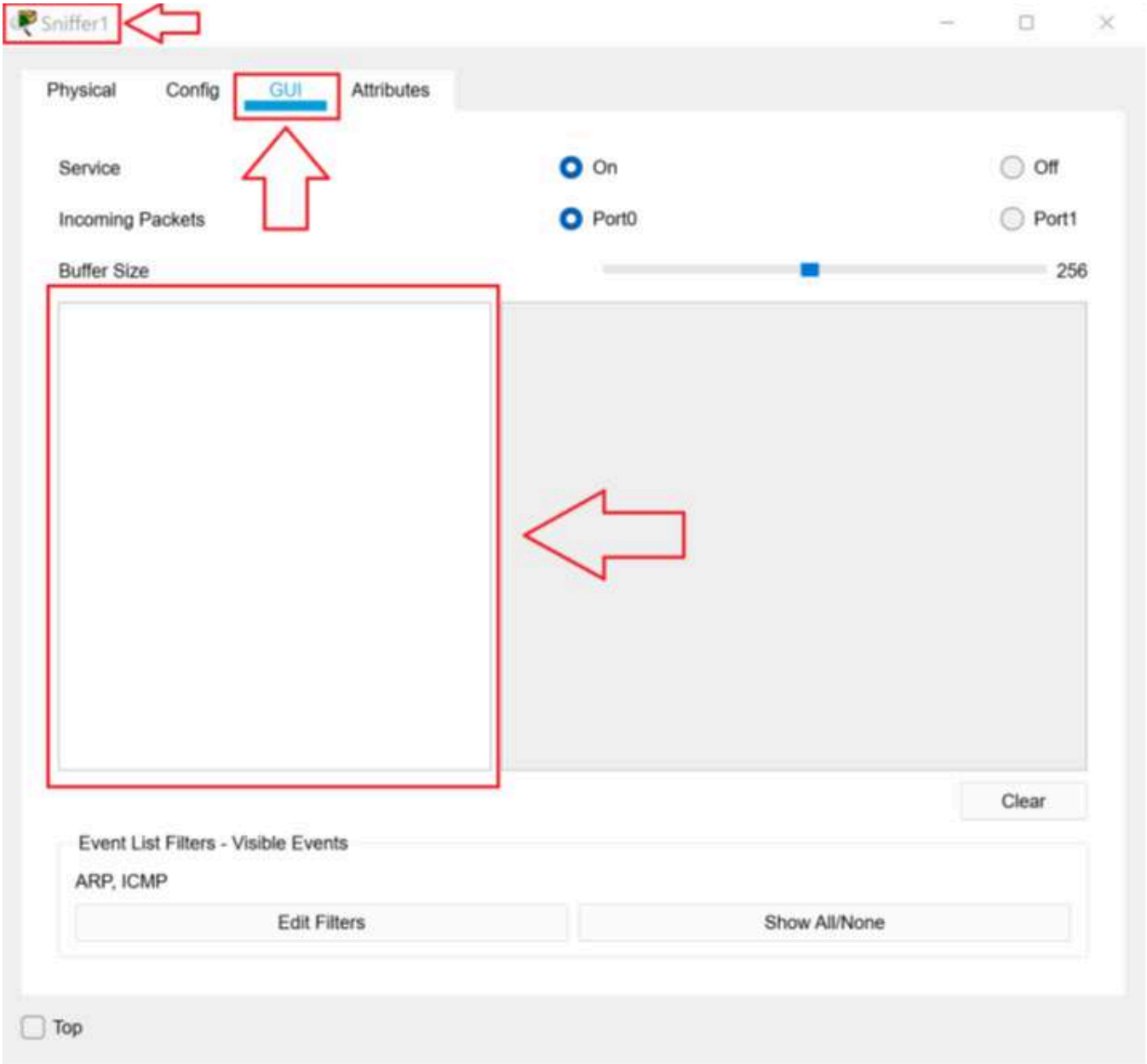
6. You should see two ARP packets and eight ICMP packets. (Don't worry if there are any other packets showing or if there are less than eight ICMP packets. If you have both ARP packets and some of the ICMP packets everything is fine.) Click on the first **ARP** packet. Scroll down in the ARP packet until you can see the **TARGET MAC** and **TARGET IP**.



7. Notice the **TARGET MAC** is all zeros. The **TARGET IP** address is 192.168.1.11. This is the ARP broadcast from 192.168.1.10 seeking the MAC address of 192.168.1.11.
8. Click on the second **ARP** packet. Scroll down in the ARP packet until you can see the **TARGET MAC** and **TARGET IP**.



9. Notice that the **TARGET MAC** is the MAC address of 192.168.1.10, which is the **TARGET IP** address. The reply from 192.168.1.11 goes directly to 192.168.1.10. PC0 knows the MAC address of 192.168.1.11 by looking at the **SOURCE MAC**.
10. Close the **Sniffer0 Properties** dialog box.
11. Click on **Sniffer1** to open the **Sniffer1 Properties** dialog box. Click the **GUI** tab. Notice that Sniffer1 did not capture any packets. The ARP broadcast was sent on the 192.168.1.0 network. The router did not pass the broadcast over to the 192.168.2.0 network because routers do not pass broadcast traffic.

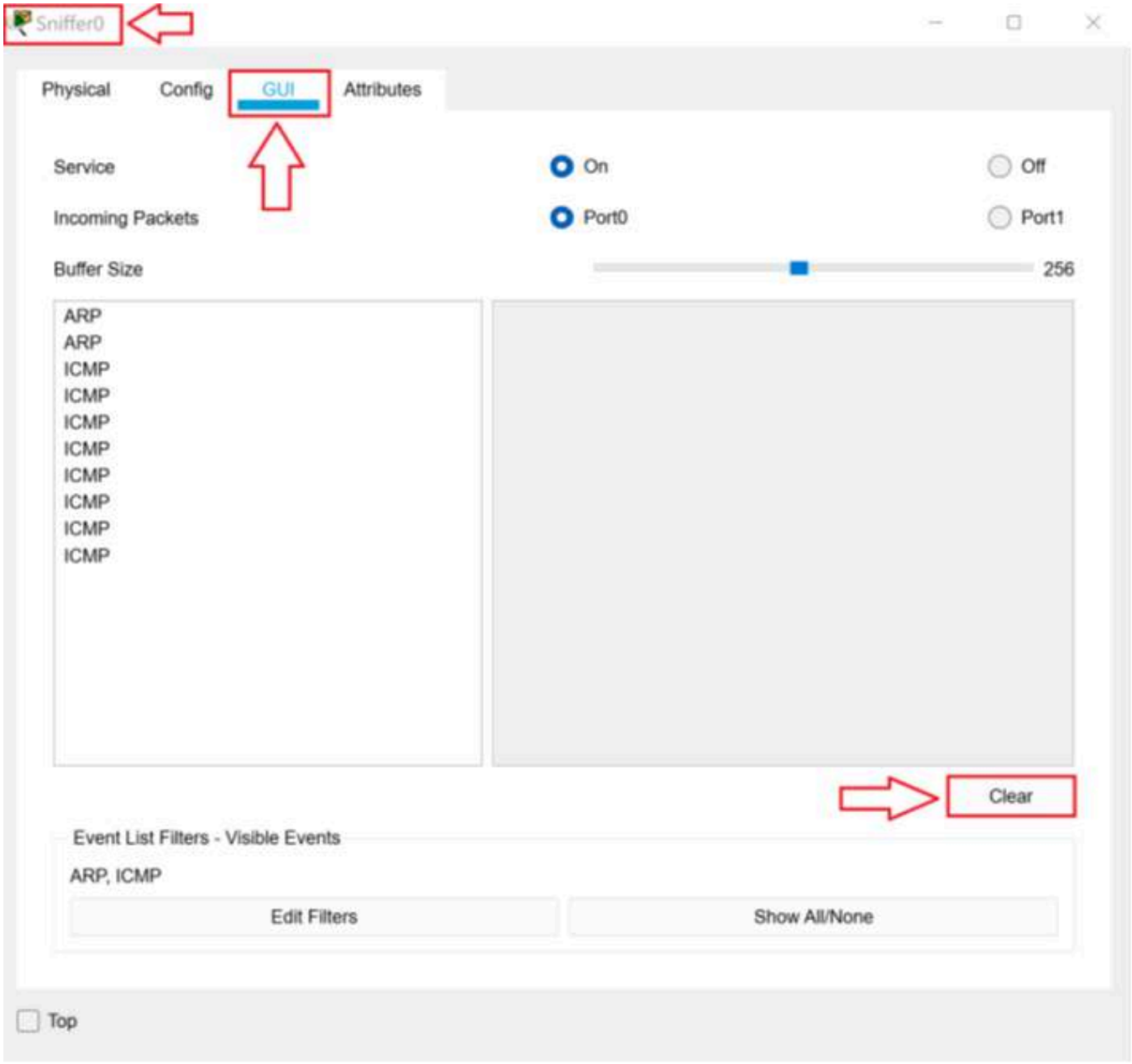


12. Close the **Sniffer1 Properties** dialog box.

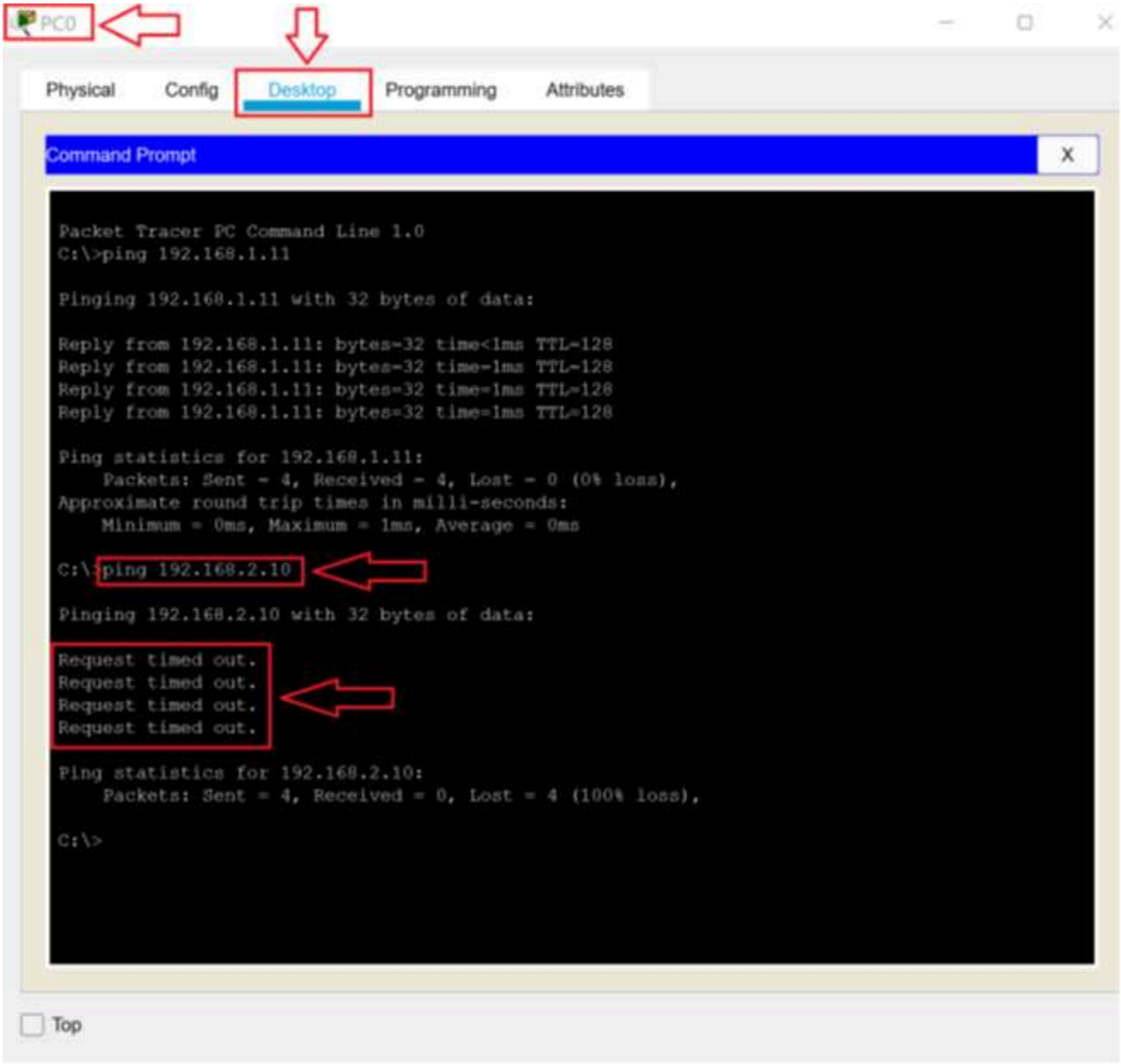
TASK B

In this task, we will look at how traffic flows between networks.

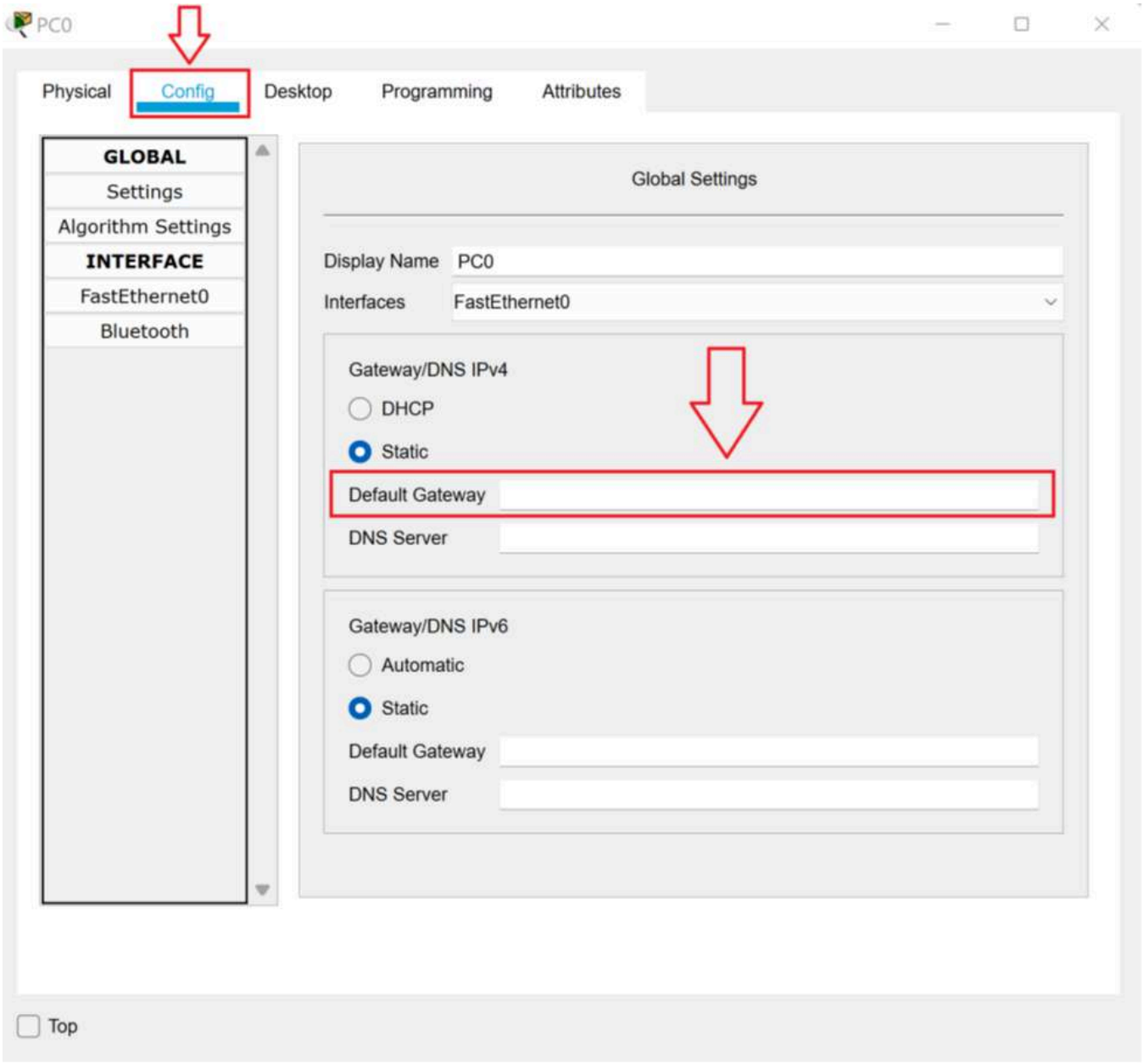
1. Click **Sniffer0**. On the **GUI** tab, click the **Clear** button to clear the buffer.



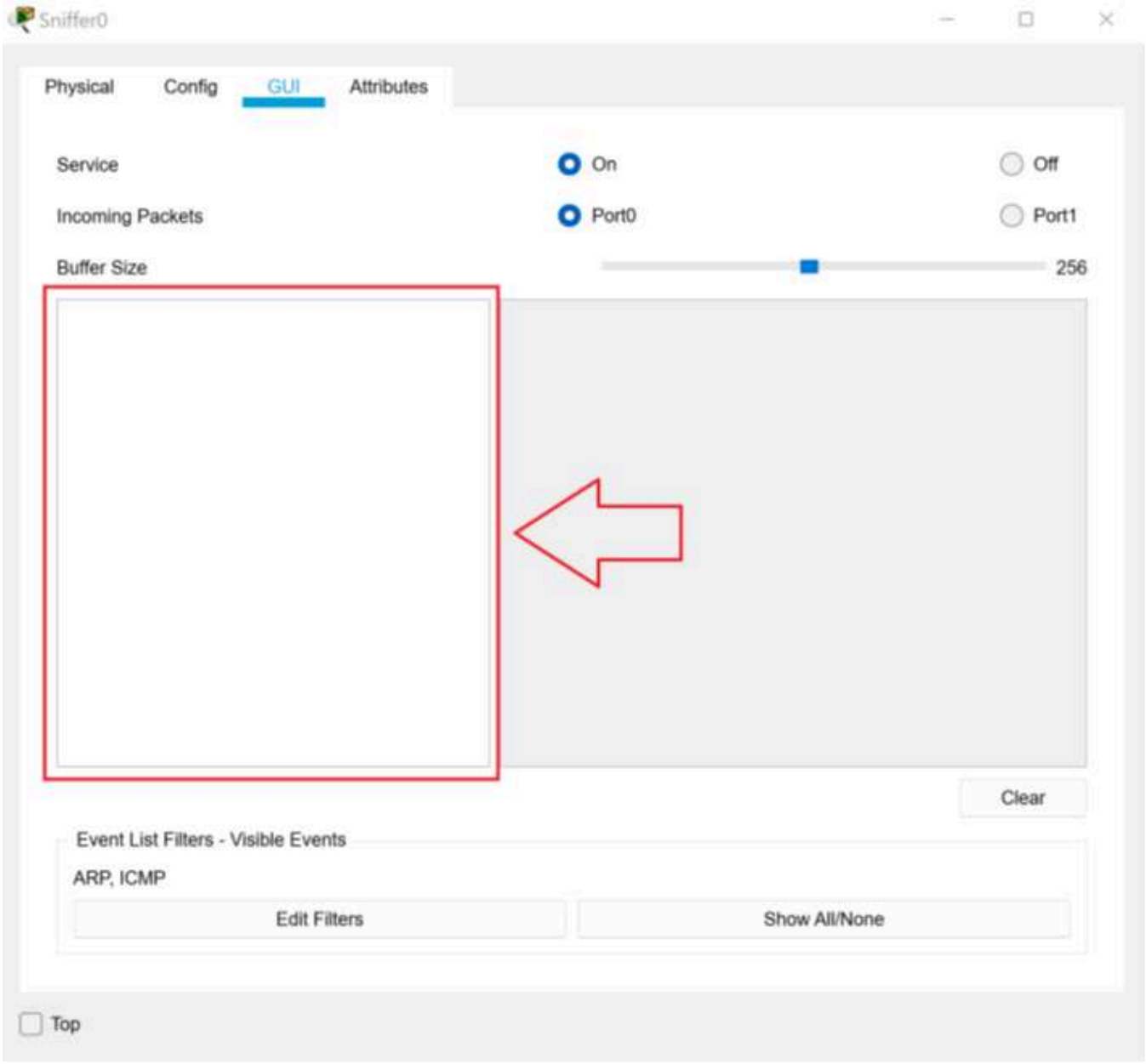
2. Close the **Sniffer0 Properties** dialog box.
3. Click **PC0**. In the **Desktop** tab, in the **Command Prompt**, type **ping 192.168.2.10** and then press Enter. Notice that 192.168.2.10 does not respond.



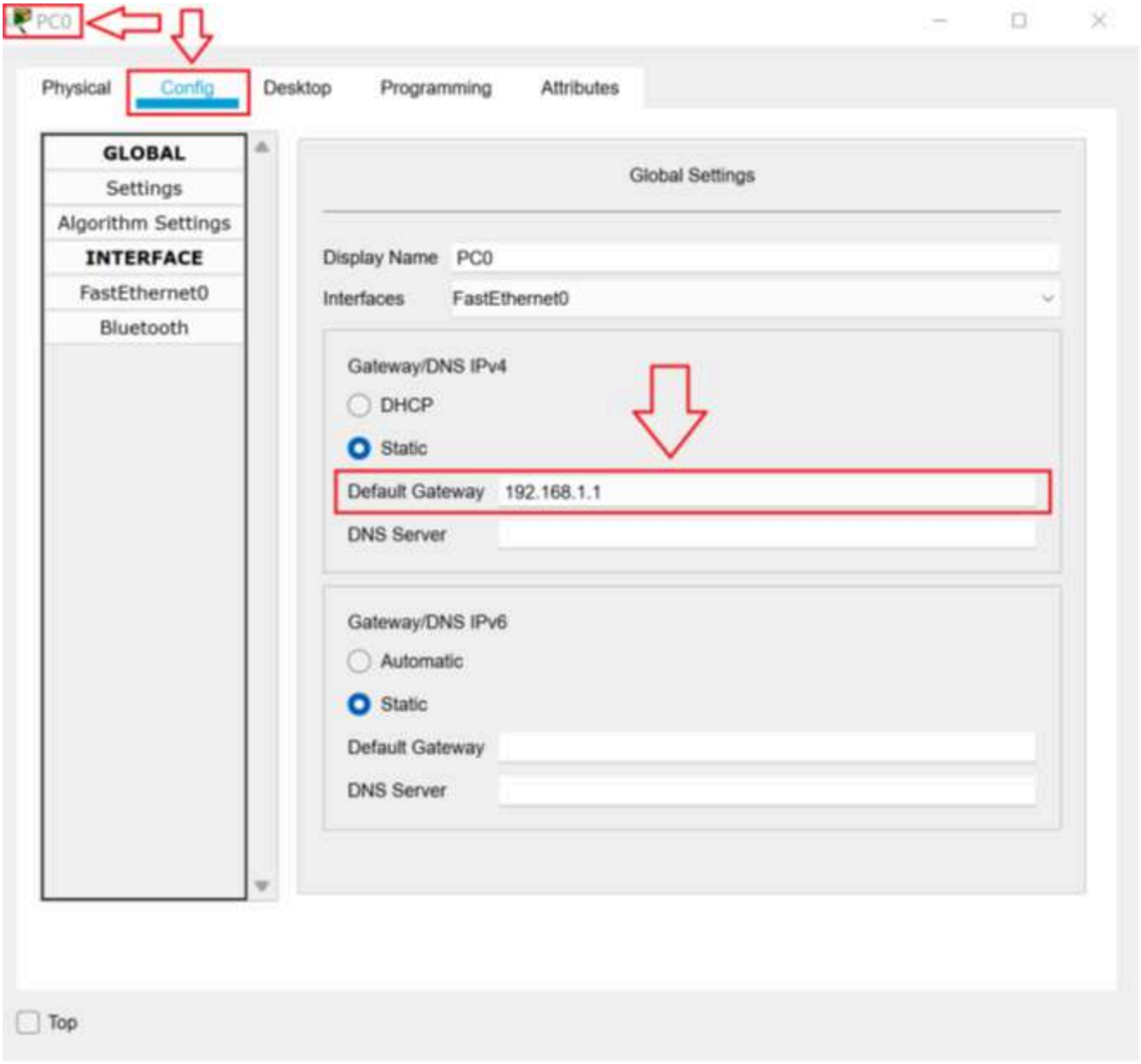
4. In the **PC0 Properties** dialog box, click the **Config** tab. Notice that PC0 does not have a default gateway setting. Since PC0 does not know the address of the router, it cannot send any packets to remote networks.



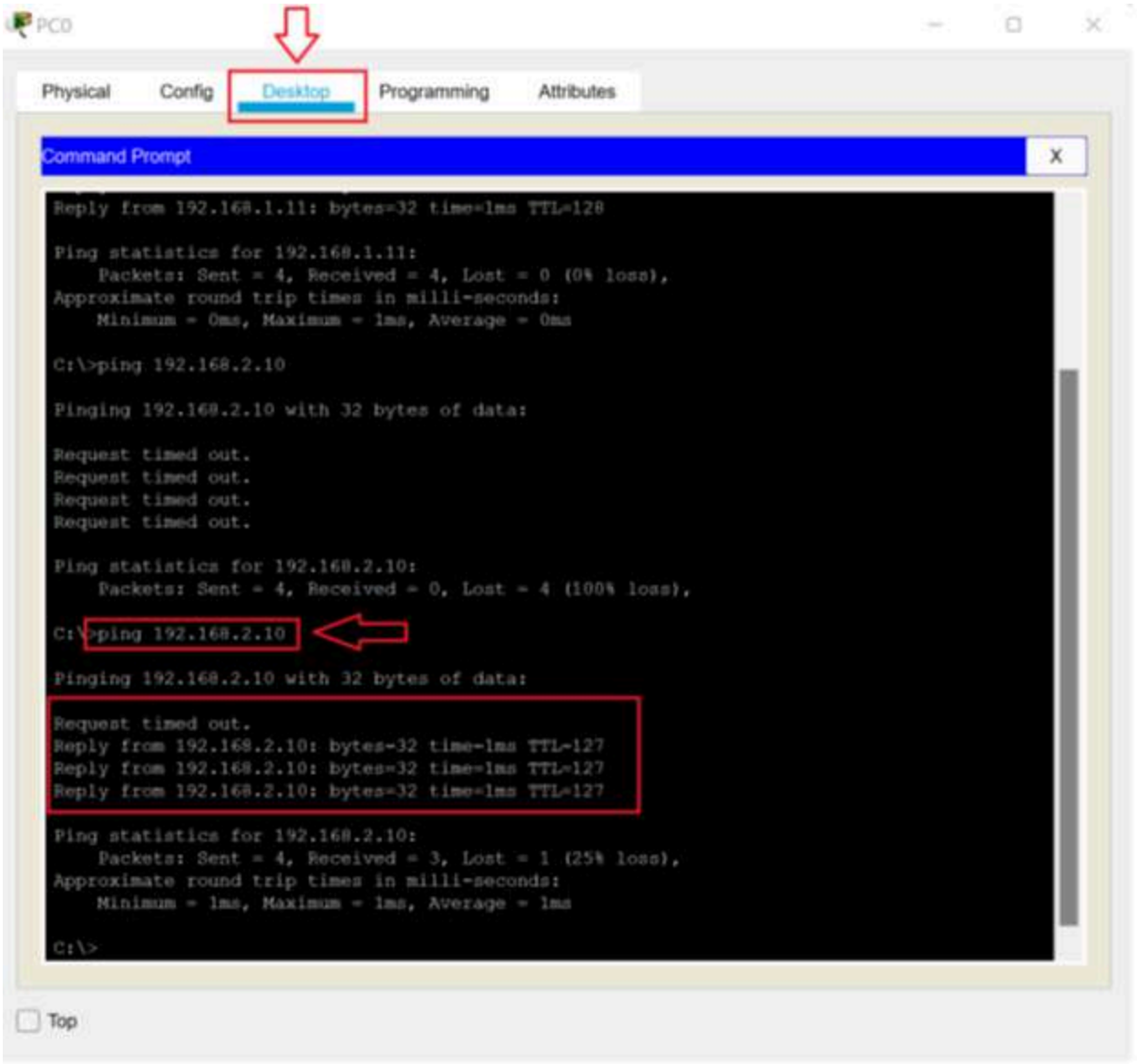
5. Close the **PC0 Properties** dialog box.
6. Click **Sniffer0**. Notice that Sniffer0 does not have any packets in its buffer. Because PC0 knows that the traffic was destined for a different network, it did not send out any ARP broadcasts for 192.168.2.10.



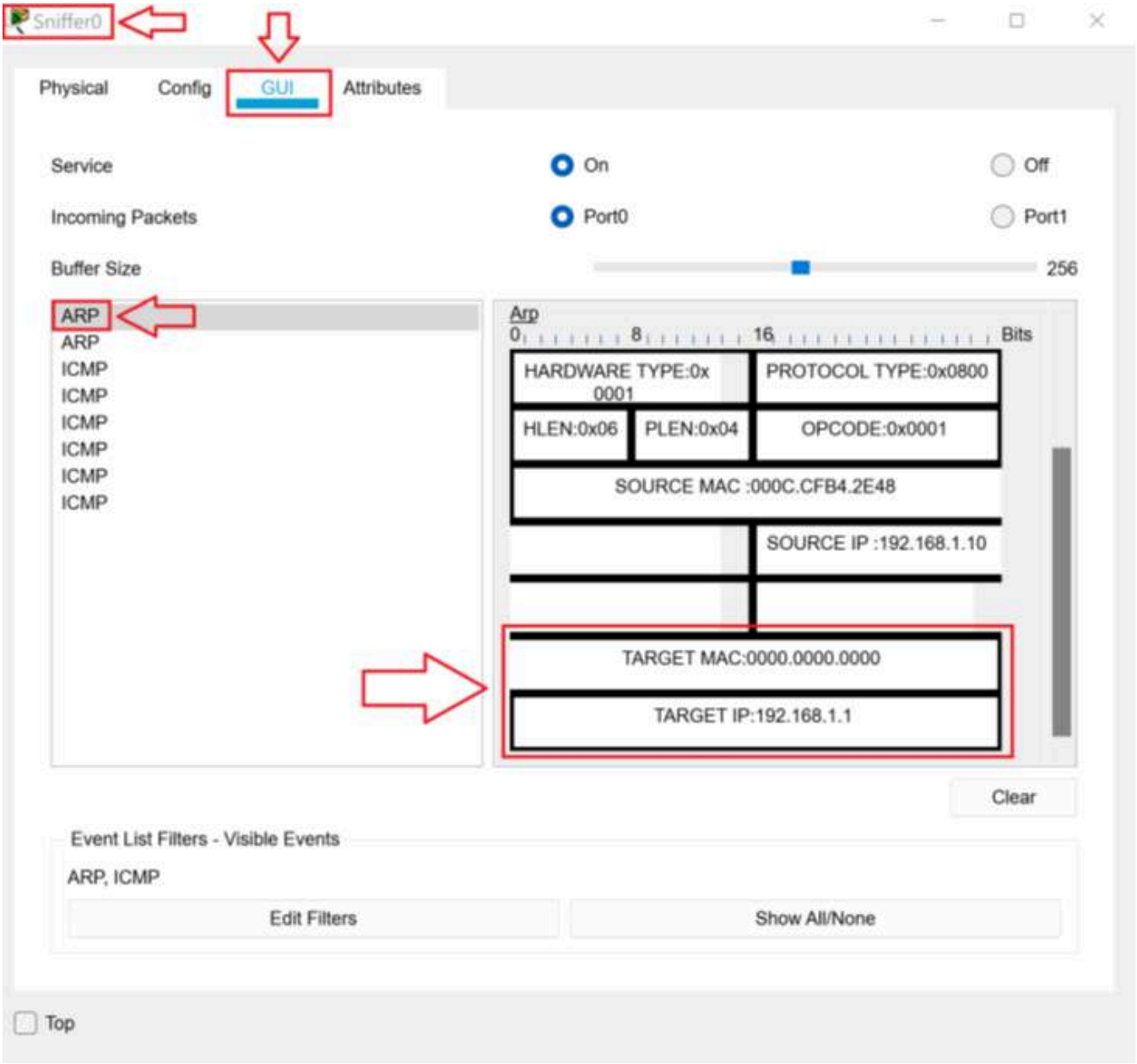
7. Close the **Sniffer0 Properties** dialog box.
8. Click **PC0**. On the **Config** tab, in the **Default Gateway** text box, type **192.168.1.1**.



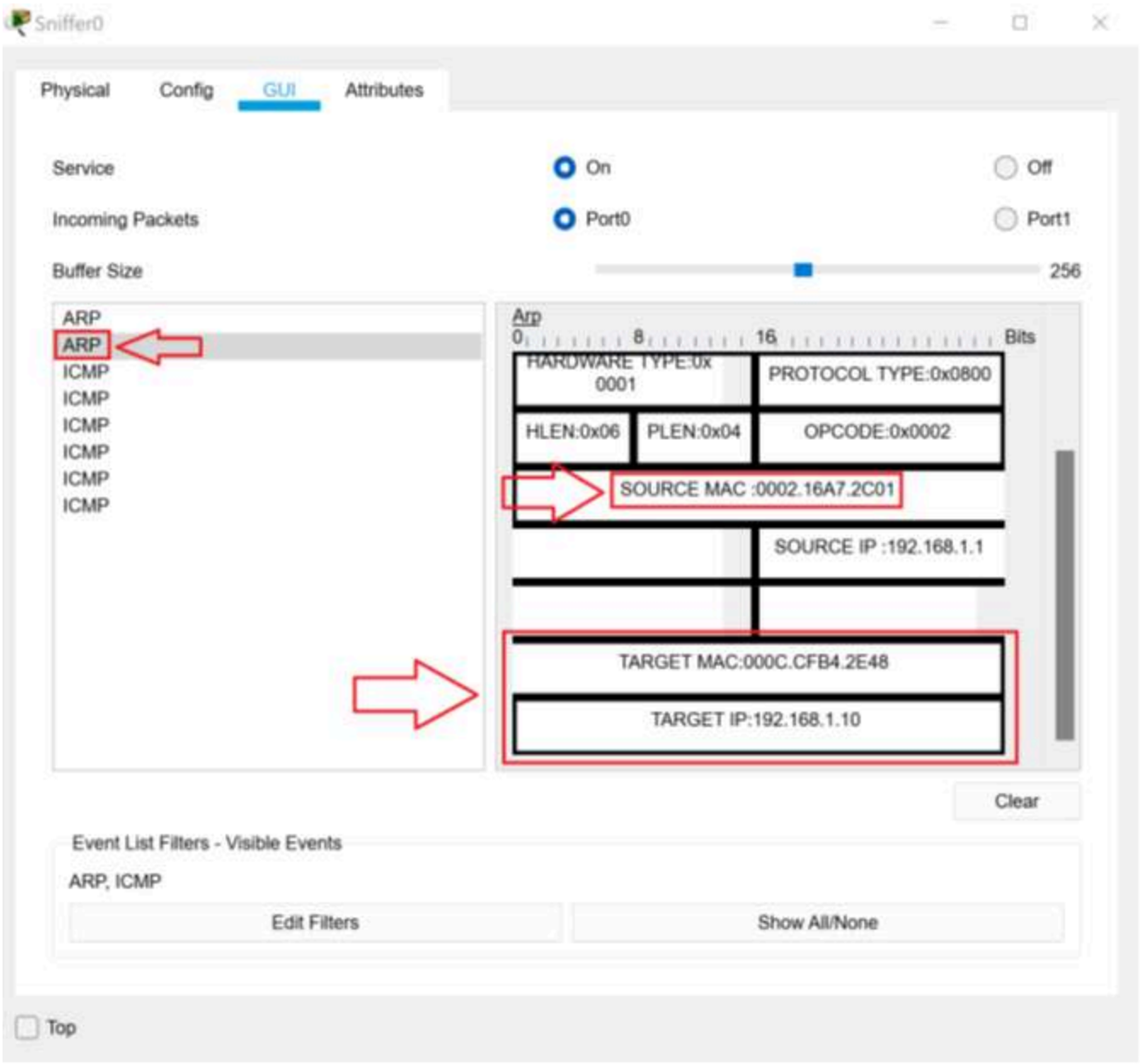
9. Click the **Desktop** tab. In the **Command Prompt**, press the **up-arrow** key on your keyboard to recall the last command (ping 192.168.2.10) and then press **Enter**. Notice that you now get a reply from 192.168.2.10. (Do not be concerned if the first one or two replies time out. It may take PC0 a few seconds to get the MAC address for the router or for the router to send and receive a reply.)



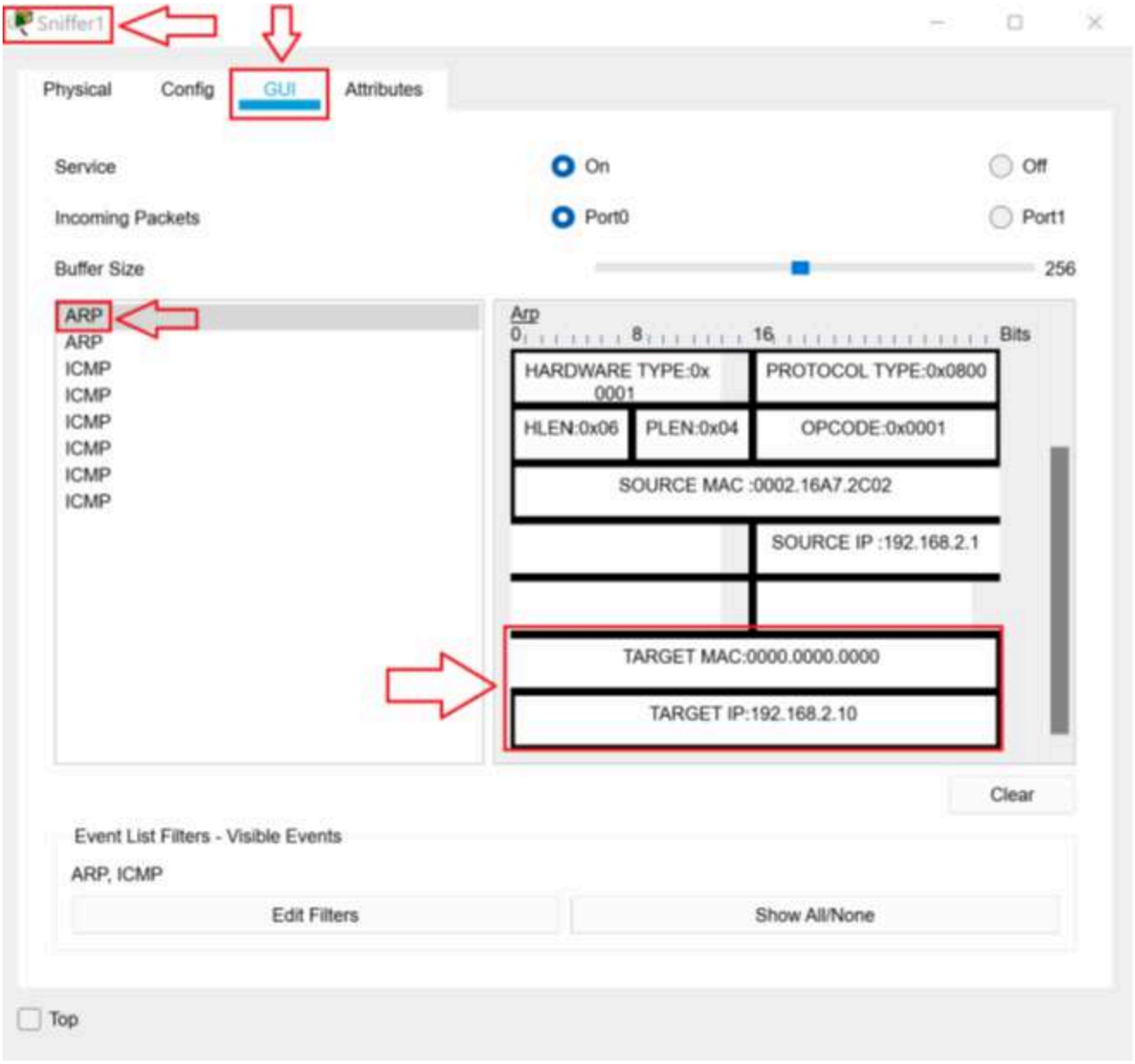
10. Close the **PC0 Properties** dialog box.
11. Click Sniffer0. On the GUI tab, notice the sniffer has captured two ARP packets and eight ICMP packets.
12. Click **Sniffer0**. On the **GUI** tab, you should see two ARP packets and eight ICMP packets. (Don't worry if there are any other packets showing or if there are less than eight ICMP packets. If you have both ARP packets and some of the ICMP packets everything is fine.) Click on the first **ARP** packet. Scroll down in the ARP packet until you can see the **TARGET MAC** and **TARGET IP**.



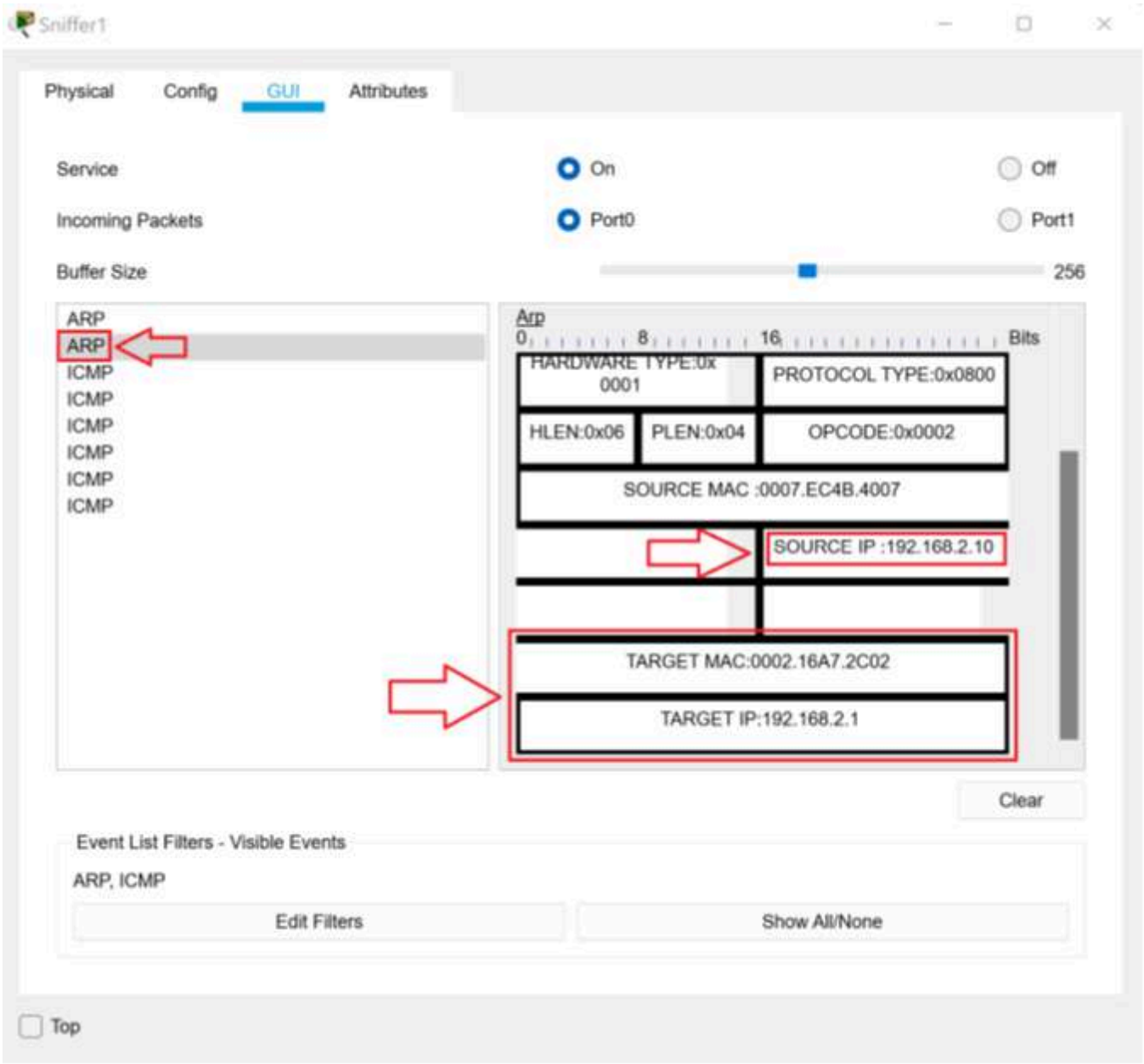
13. Notice that the **TARGET MAC** address is all zeros. The **TARGET IP** address is 192.168.1.1. Since PC0 needs to send the packets to the 192.168.2.0 network, this is the broadcast to find the MAC address of the router.
14. Click on the second **ARP** packet. Scroll down in the ARP packet until you can see the **TARGET MAC** and **TARGET IP**.



15. Notice that the **TARGET MAC** is the MAC address of 192.168.1.10, which is the **TARGET IP** address. The reply from 192.168.1.1 goes directly to 192.168.1.10. PC0 knows the MAC address of 192.168.1.1 by looking at the **SOURCE MAC**.
16. Close the **Sniffer0 Properties** dialog box. Click **Sniffer1**. On the **GUI** tab, you should see two ARP packets and eight ICMP packets. (Don't worry if there are any other packets showing or if there are less than eight ICMP packets. If you have both ARP packets and some of the ICMP packets everything is fine.) Click on the first **ARP** packet. Scroll down in the ARP packet until you can see the **TARGET MAC** and **TARGET IP**.



17. Notice that the **TARGET MAC** address is all zeros. The **TARGET IP** address is 192.168.2.10. At this point, the ping from the 192.168.1.0 network has arrived at the router. The router is sending an ARP broadcast to get the MAC address of 192.168.2.10.
18. Click on the second **ARP** packet. Scroll down in the ARP packet until you can see the **TARGET MAC** and **TARGET IP**.



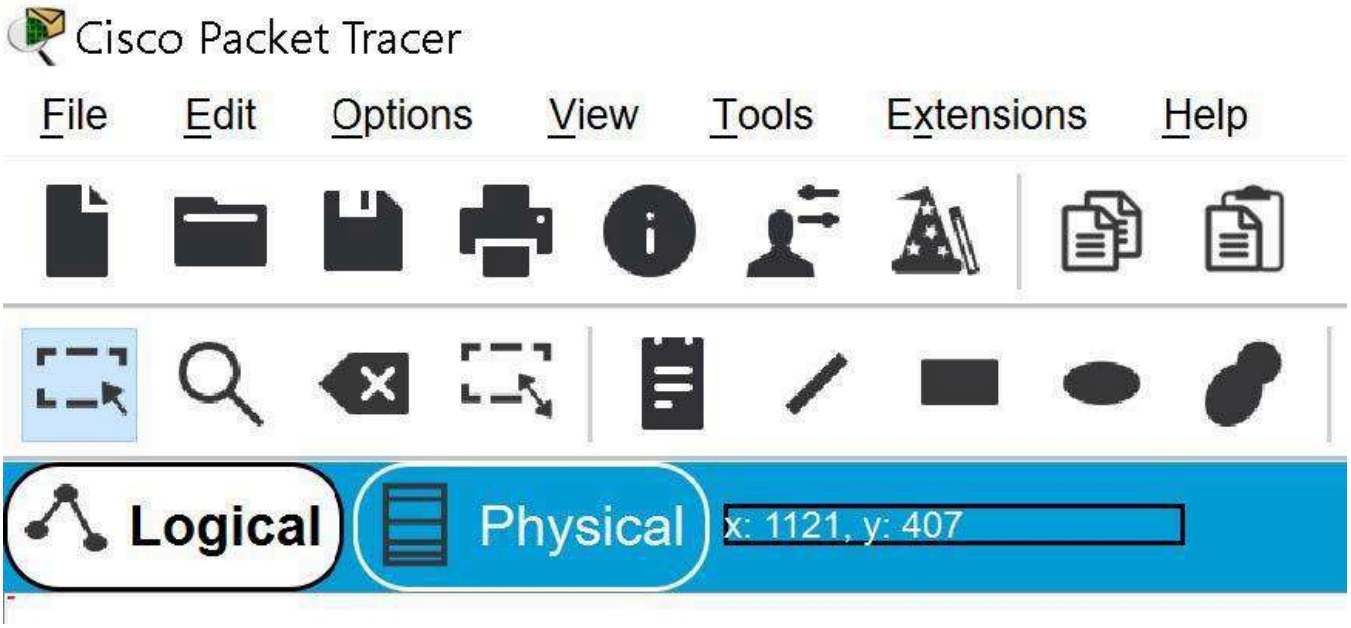
19. Notice that the **TARGET MAC** is the MAC address of 192.168.2.1, which is the **TARGET IP** address. The reply from 192.168.2.10 goes directly to 192.168.2.1. The router knows the MAC address of PC2 by looking at the **SOURCE MAC**. Close the **Sniffer1** Properties dialog box.
20. Close the **3.4.3 Lab File** file.

Configure Wireless

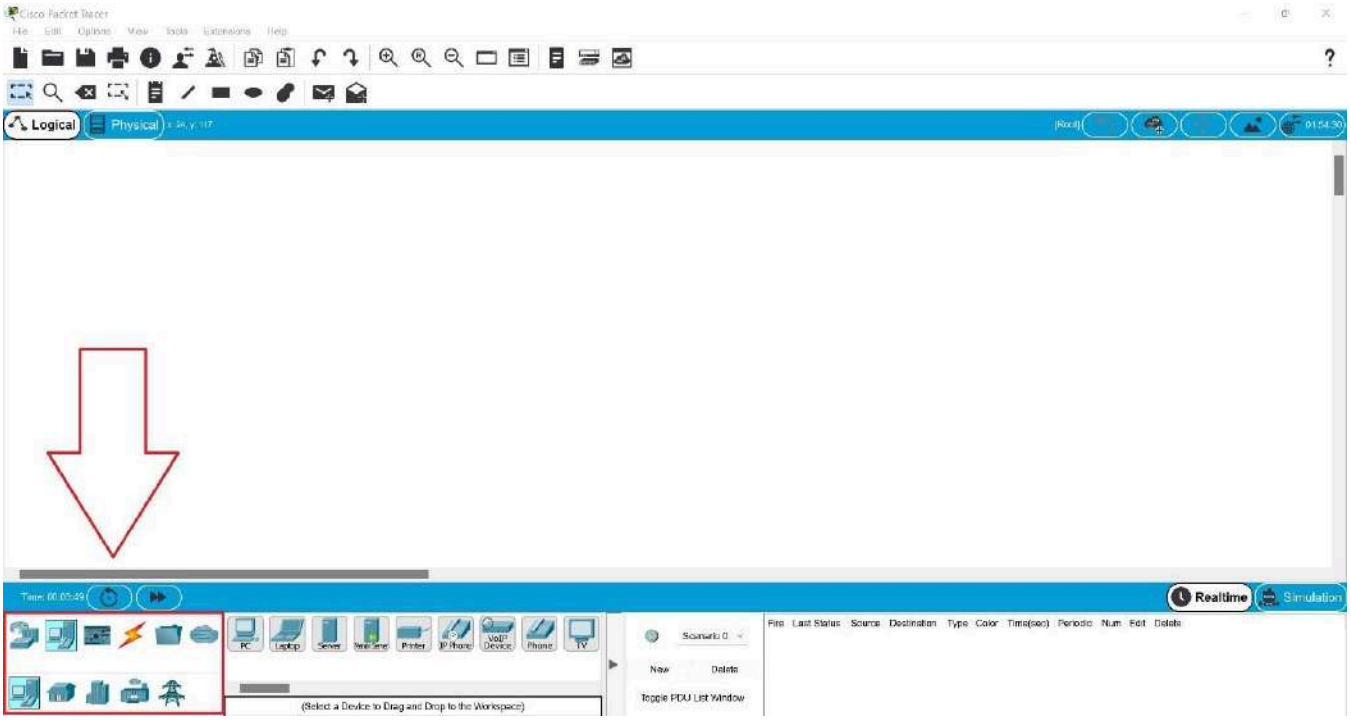
In this lab, we will set up a small wireless network.

TASK A

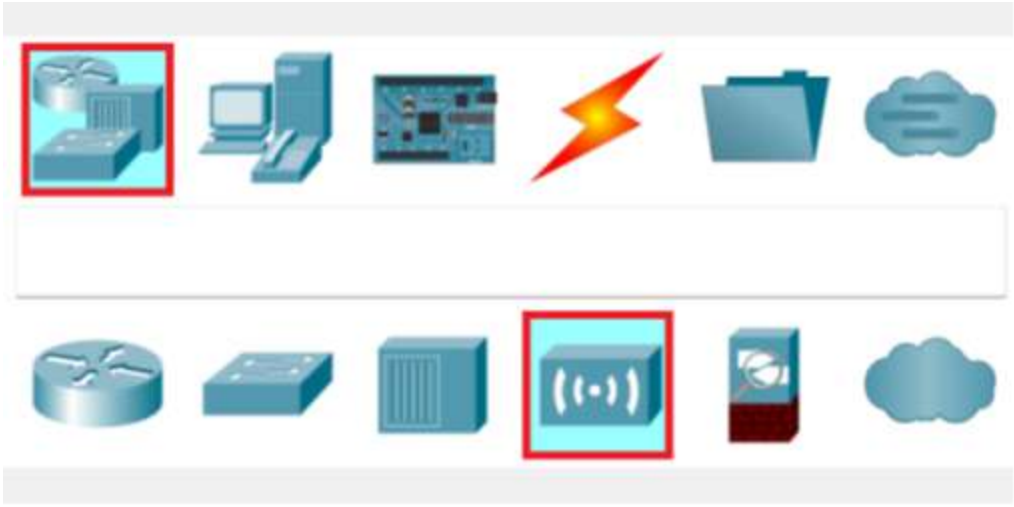
1. Open **Packet Tracer**.
2. In the upper left corner, make sure that the **Logical** button is white.



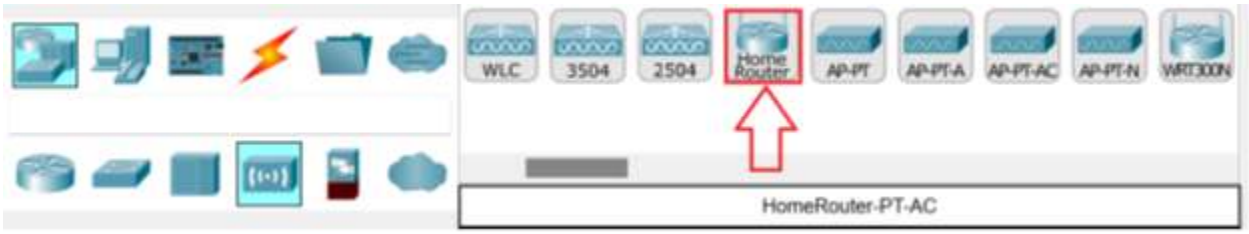
3. In the bottom left corner of the screen, locate the **toolbox area**.



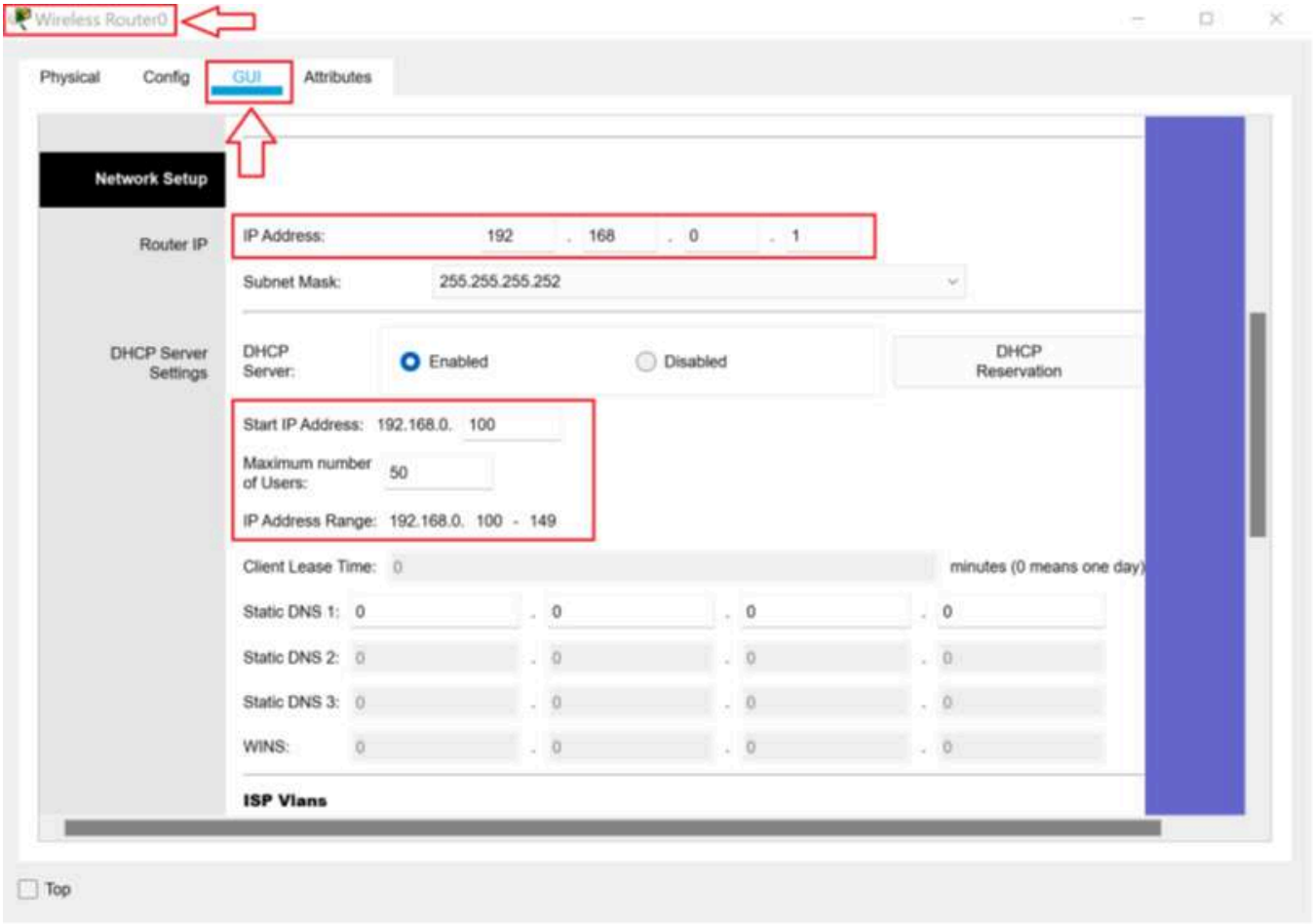
4. In the **toolbox area**, select **Network Devices** and then select **Wireless Devices**.



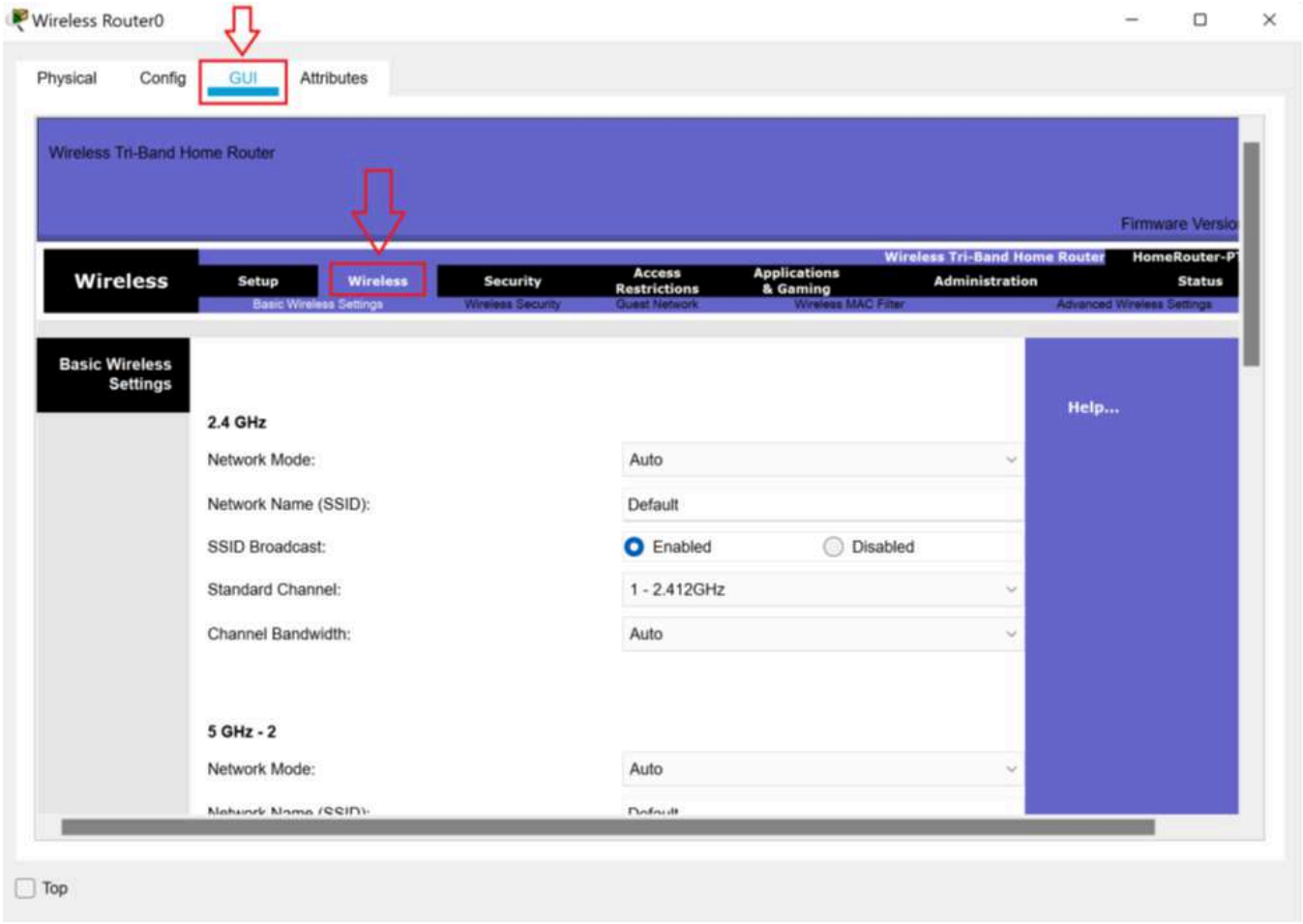
5. Drag a **Home Router** onto the logical diagram.



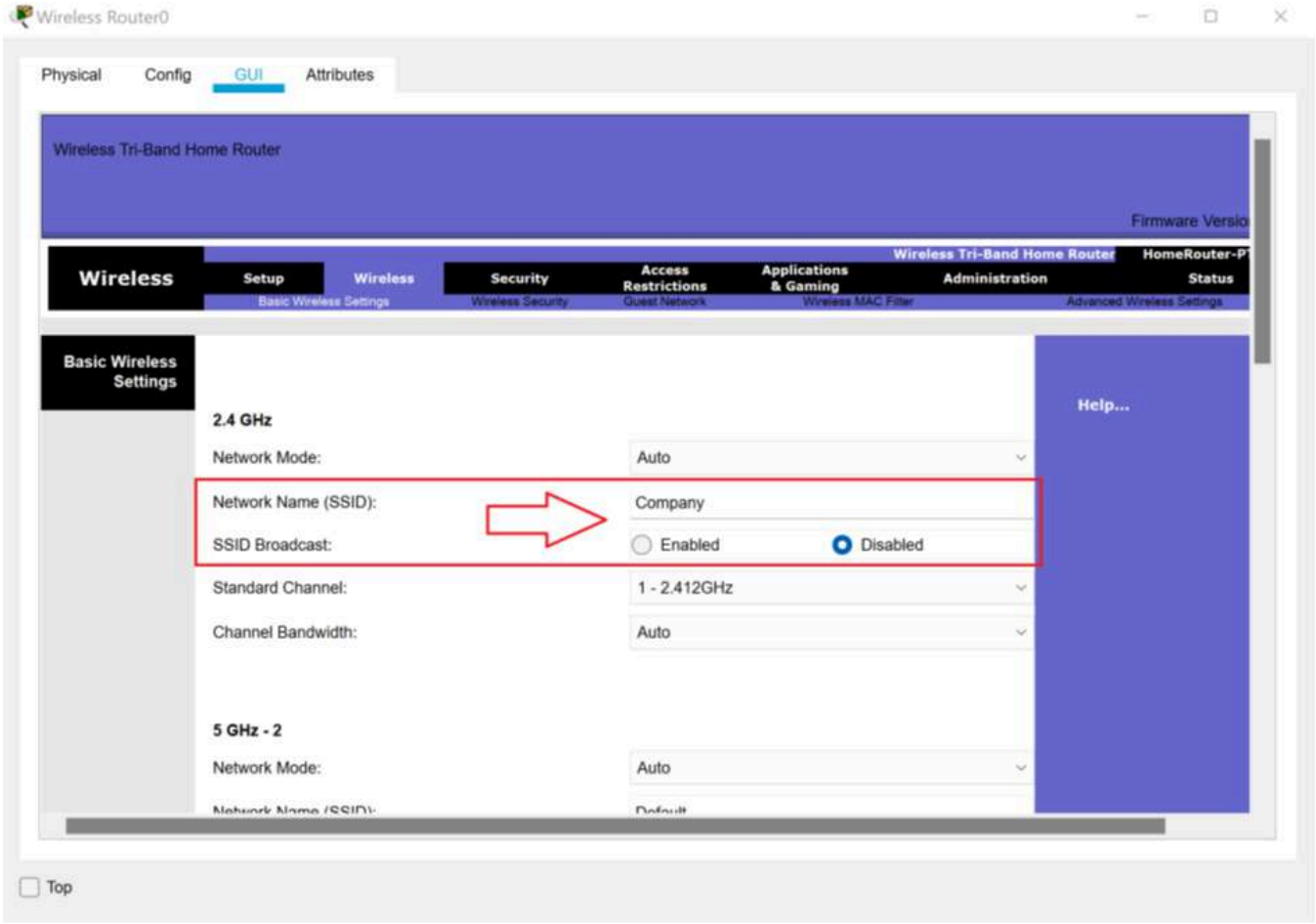
6. Click **Wireless Router0** to open the **Wireless Router0 Properties** dialog box. Then click the **GUI** tab. Scroll down to the **Network Setup** section.



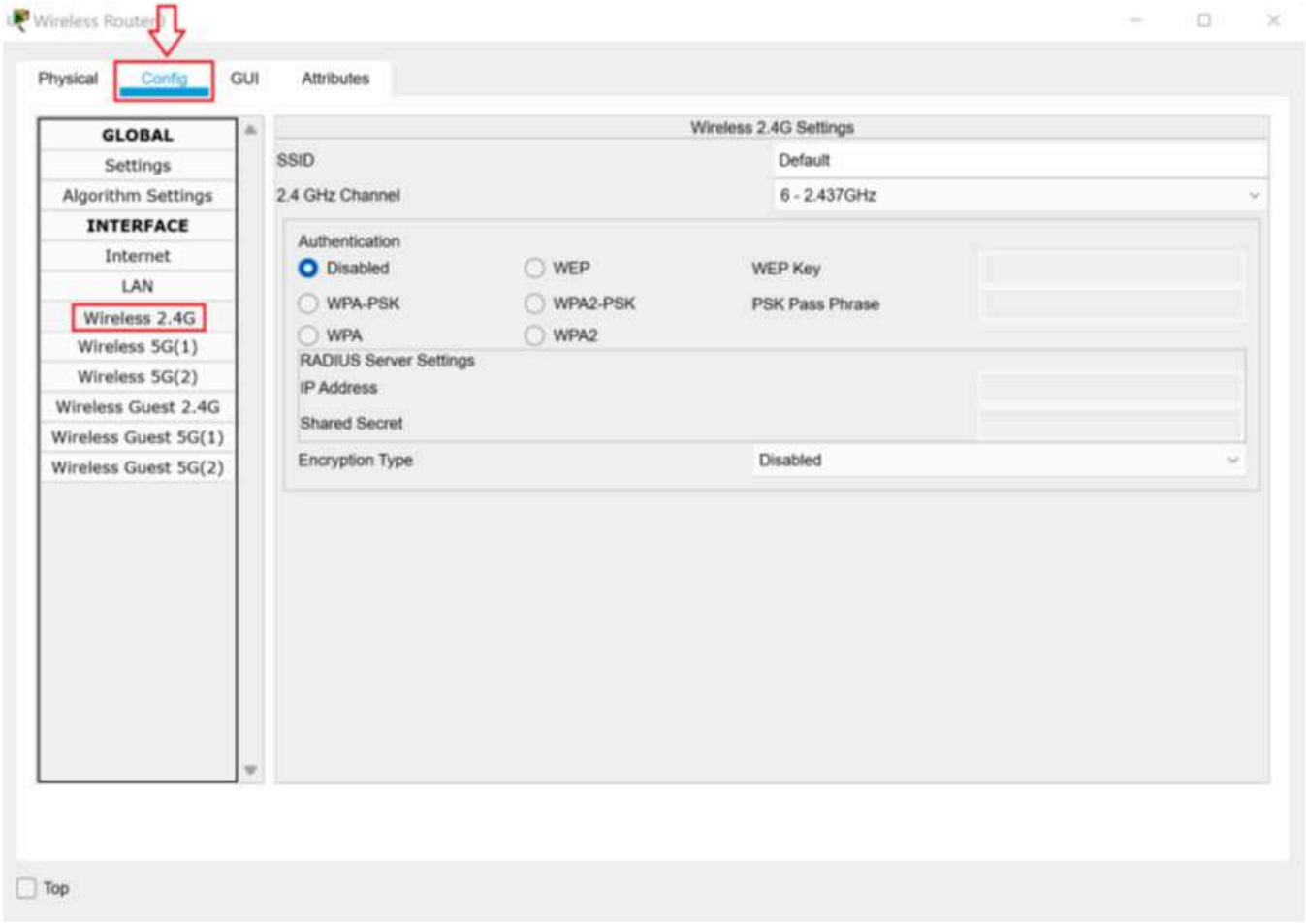
7. Notice the router's IP address is 192.168.0.1. The router will function as a DHCP server. It can give out 50 IP addresses starting with 192.168.0.100.
8. In the **GUI** tab, click the **Wireless** menu option.



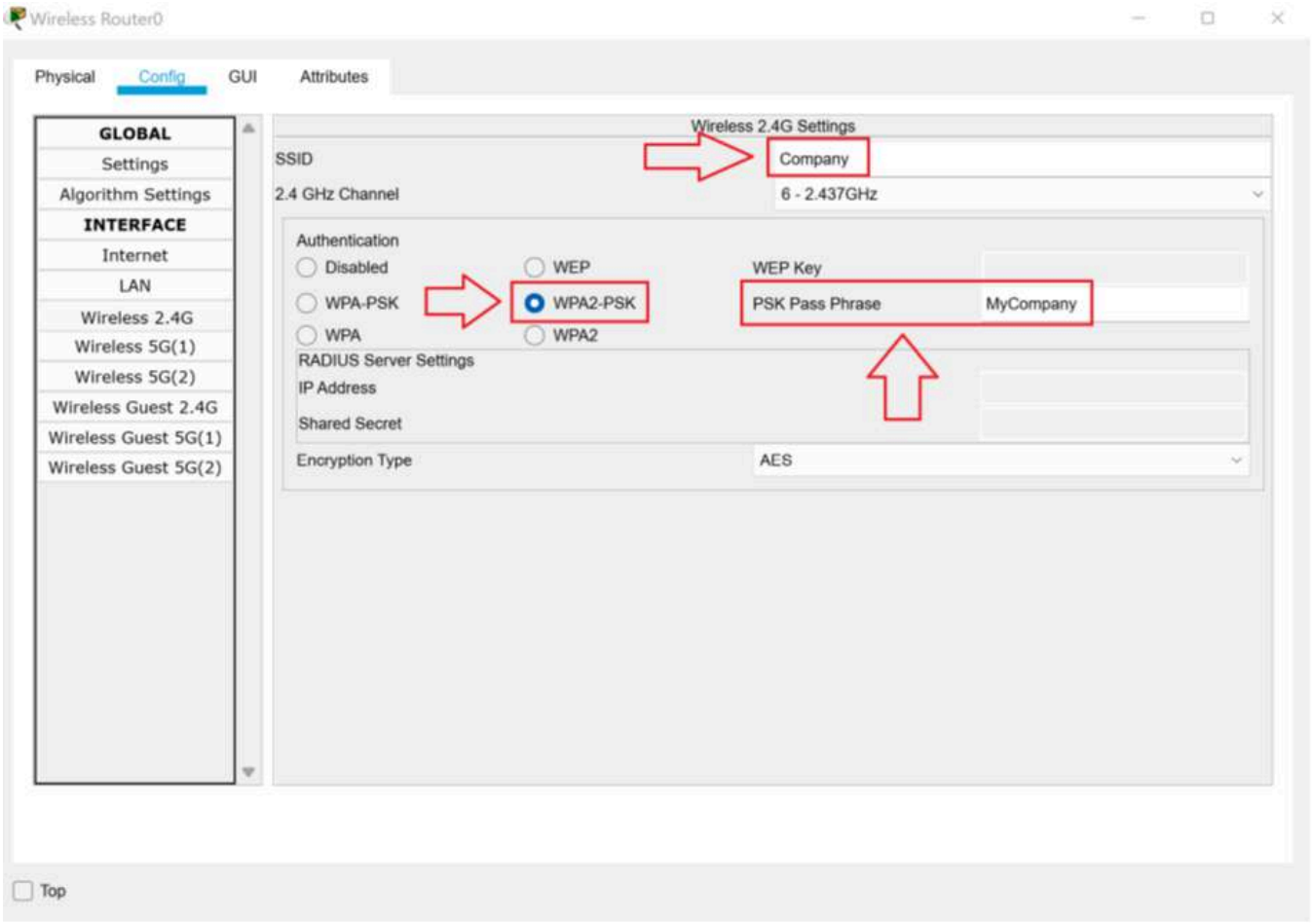
9. The wireless clients will connect to the wireless routing using the 2.4 GHz band. Change the **Network Name (SSID)** to **Company** and set **SSID Broadcast** to **Disabled**.



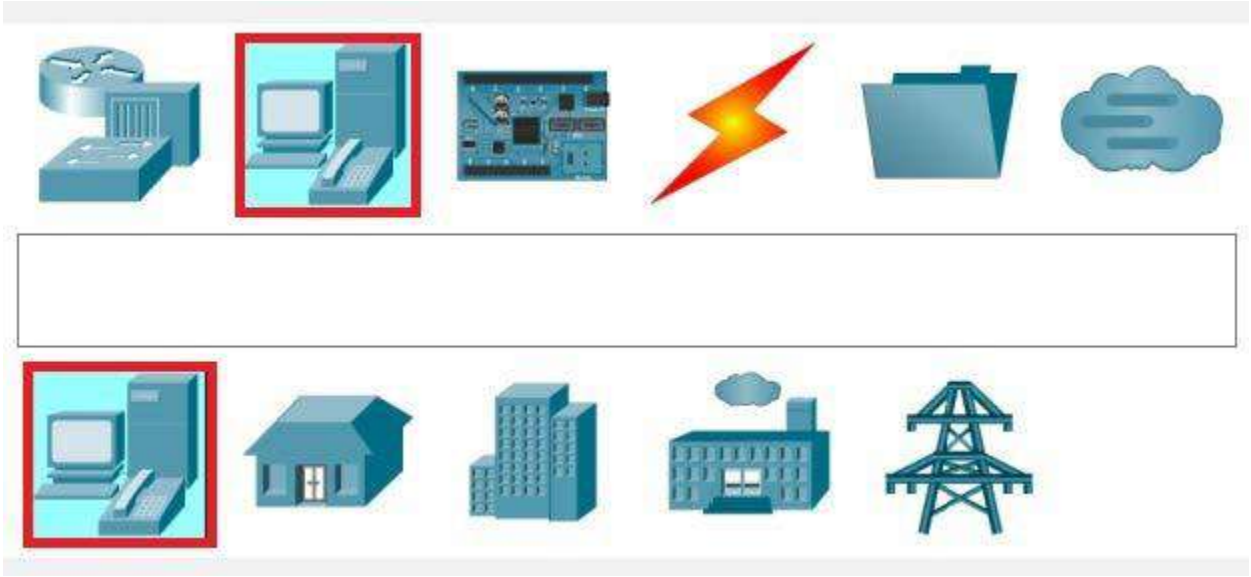
10. In the **Wireless Router0 Properties** dialog box, click the **Config** tab. In the **Interface** menu, click **Wireless 2.4G**.



11. In the Authentication section, select the **WPA2-PSK** radio button. Set the **PSK Pass Phrase** to **MyCompany**.

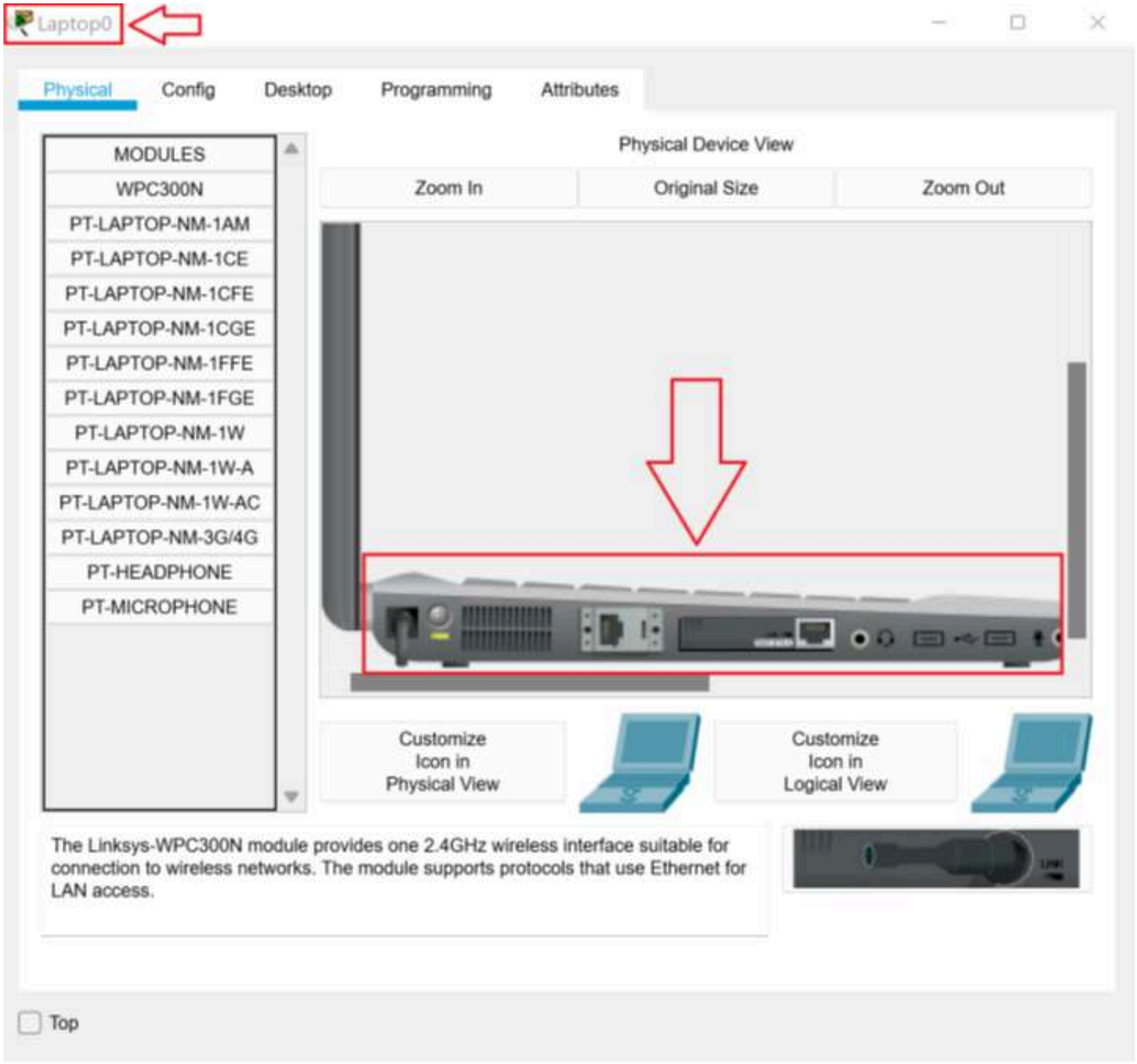


12. Close the **Wireless Router0 Properties** dialog box.
13. In the **toolbox area**, select **End Devices** and then select **End Devices**.



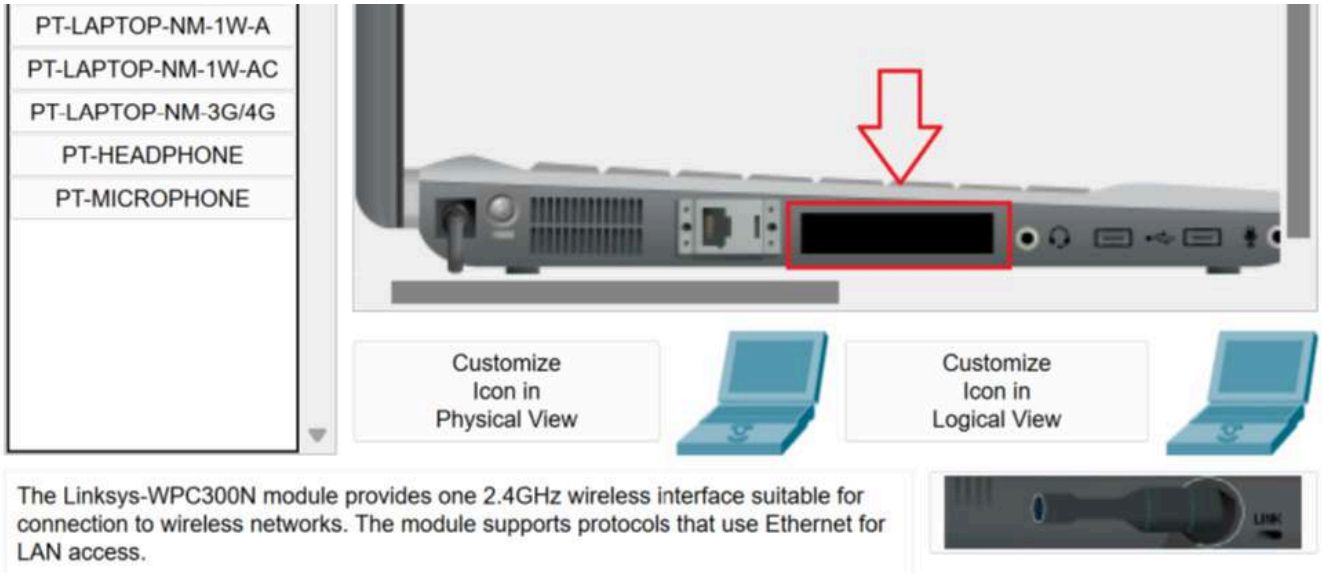
14. Drag a **Laptop** onto the logical diagram.

15. Click on **Laptop0** to open the **Laptop0 Properties** dialog box. Scroll down so you can see the side of the laptop.

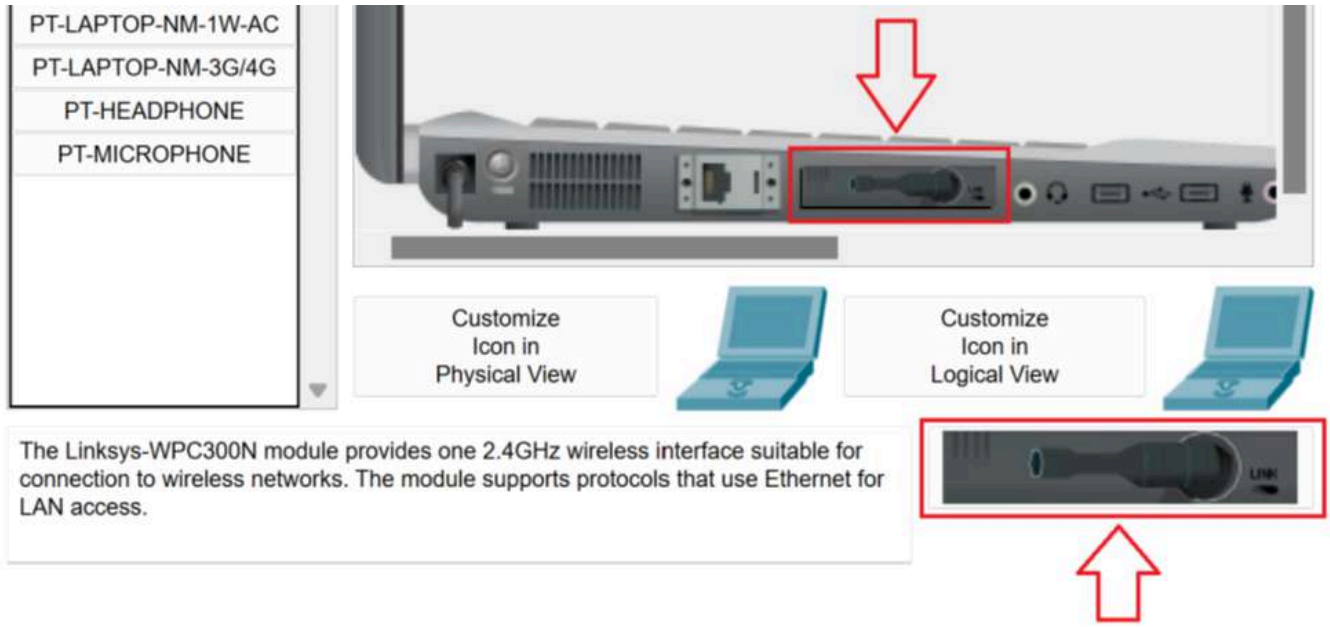


16. The laptop comes configured with a wired network card. To use the wireless, we need to change out the network card. That can only be done when the laptop is off. To power off the laptop, click the power button. The power light will change from green to gray.

17. Drag the network card from the laptop to the modules section. The network card area will be empty.

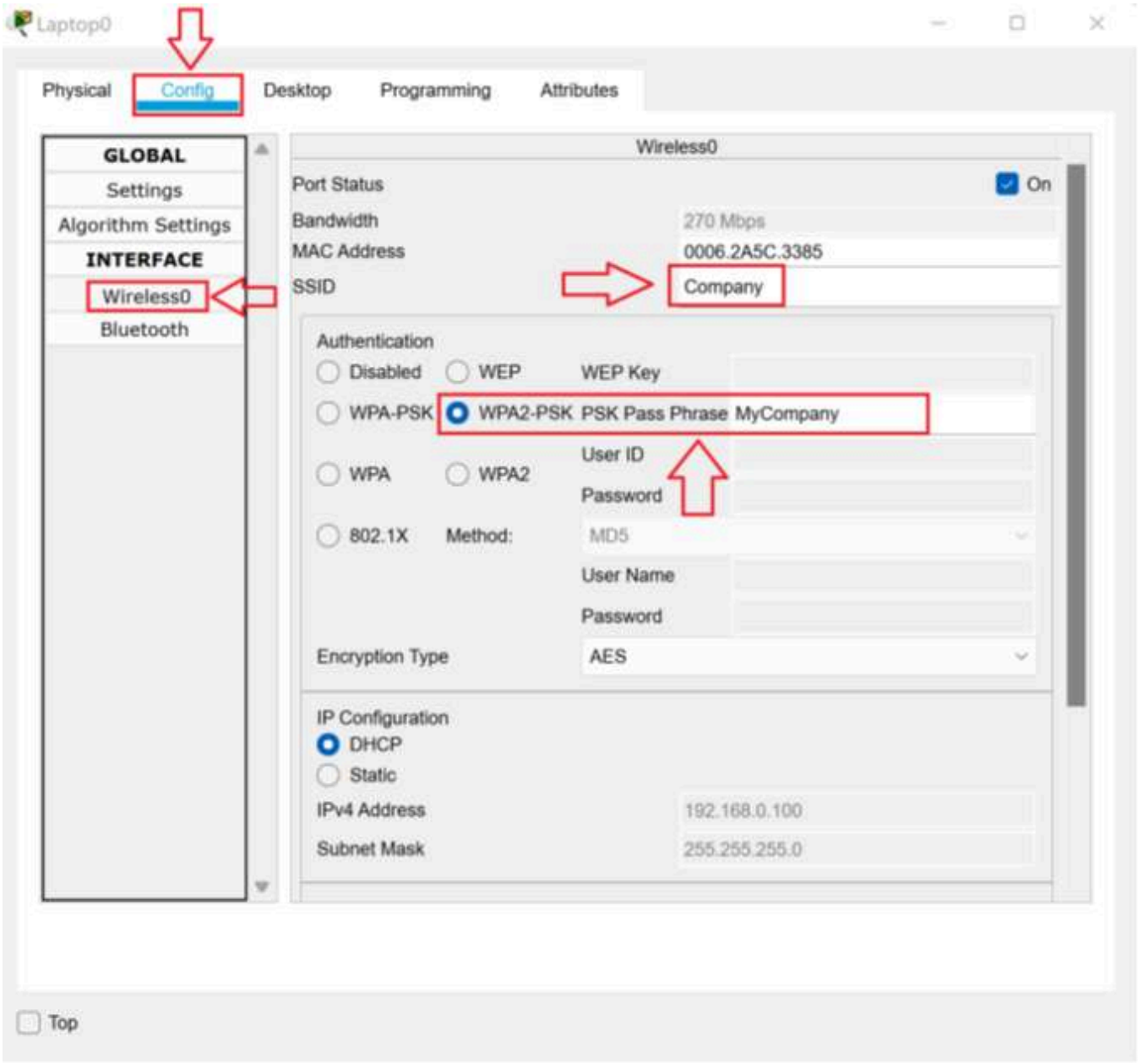


18. From the bottom right of the **Laptop0 Properties** dialog box, drag the wireless adapter to the empty network card slot.

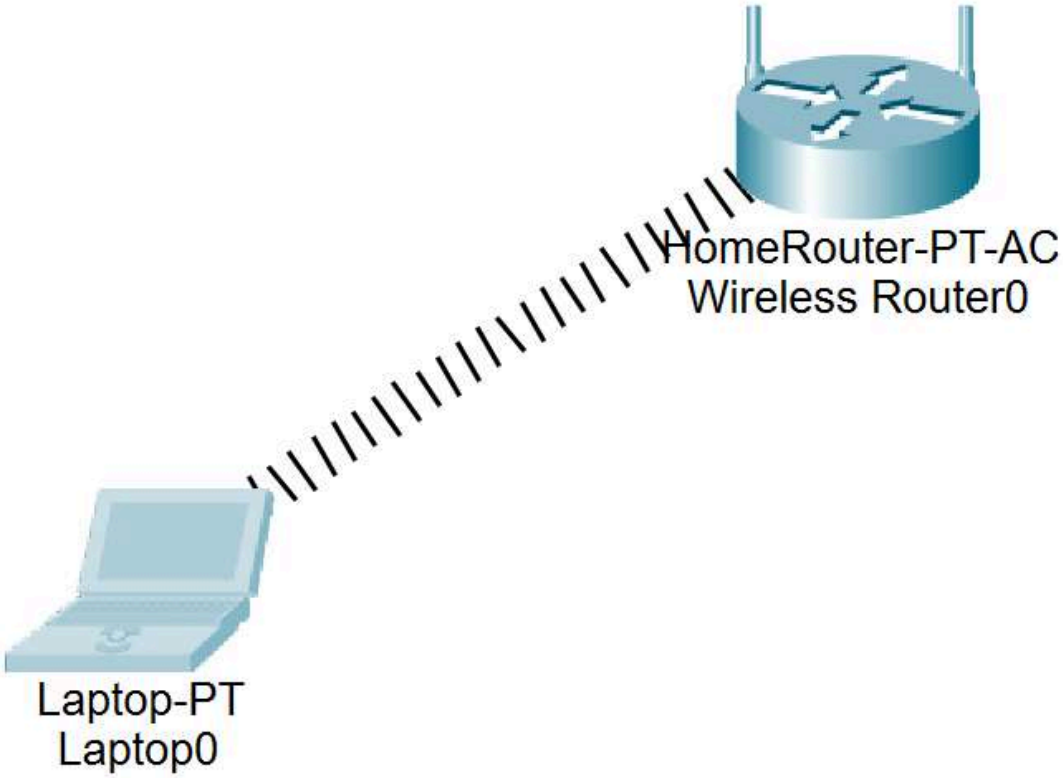


19. Click the power button to power on the laptop. The power light will change from gray to green.

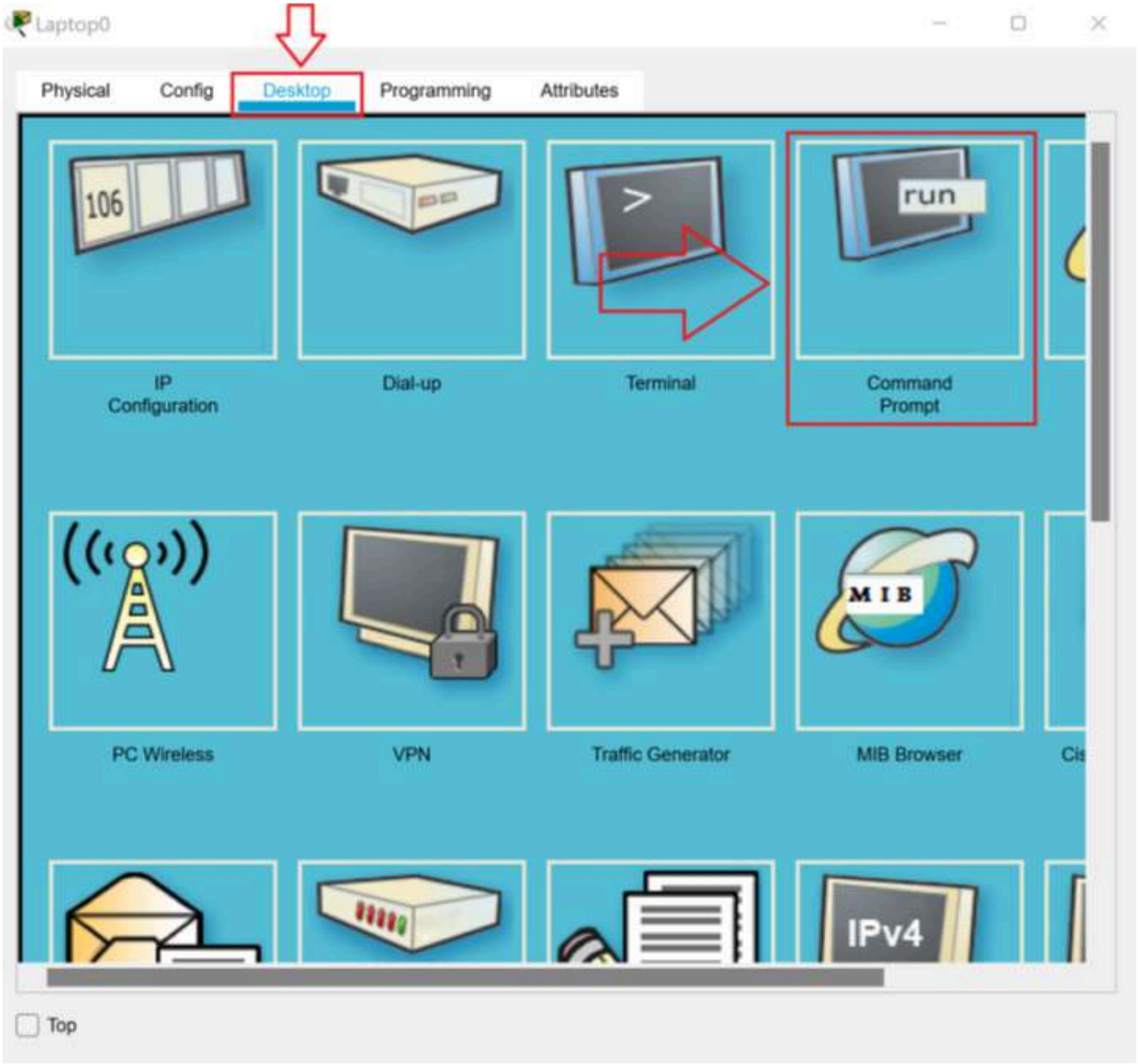
20. In the **Laptop0 Properties** dialog box, click the **Config** tab. In the **Interface** menu, click **Wireless0**. Change the SSID to **Company**. Select the **WPA2-PSK** radio button. In the **PSK Pass Phrase** text box, type **MyCompany**.



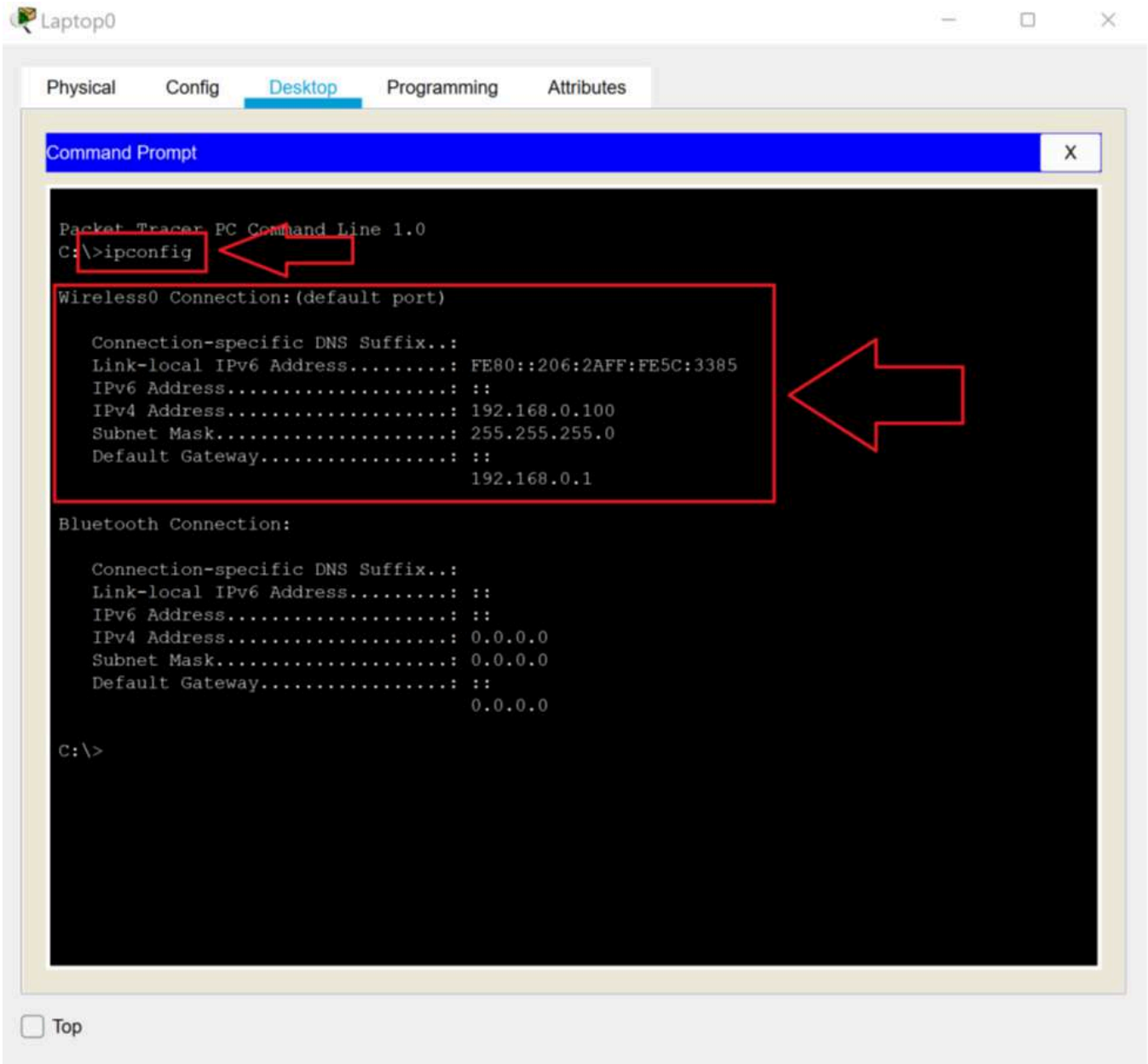
21. In the logical diagram, the laptop will show a wireless connection with the router.



22. In the **Laptop0 Properties** dialog box, click the **Desktop** tab and then click the **Command Prompt** icon.



23. In the **Command Prompt**, type **ipconfig** and press **Enter**. Notice the laptop picked up the first IP address in the DHCP address range from the router.



24. Close **Packet Tracer**. It is not necessary to save the changes.

IPV4 Addressing

Background

In 1957, the United States government formed the Advanced Research Projects Agency (ARPA). Their goal was to make the US a leader in military science and technology. In 1962, there was concern about nuclear war. If bombed, the US needed to be able to return fire.

In 1962, the US Air Force conducted a study on how to keep control of missiles and bombers after a nuclear attack. The report recommended creating a decentralized military research network. In 1969, as a solution to this problem, ARPA launched ARPANET. ARPANET was a wired network that originally connected very few nodes. This is the network that became the backbone of the Internet.

ARPANET, with its wired connections, was stable. But packet radio and satellite networks carried data over longer distances than wire. They also used different, incompatible protocols. To connect these dissimilar networks, ARPANET needed a universal protocol to support “internet working.” In 1973, Vinton Cerf and Bob Kahn, while holed up in a hotel room for two days, brainstormed the concept of TCP/IP.

TCP/IP was inspired by an analogy to the postal system.

Imagine a letter written in English but sent to a country, like China, which doesn’t use the same language (alphabet). When the letter arrives at the first post office which doesn’t use that alphabet (language), how can the post office understand the address enough to deliver it?

One solution might be to put the letter inside another envelope. The outer envelope could be addressed using the local language. As the letter travels between countries, each post office can remove the letter from the outer envelope. Then they can put it inside another envelope written in the local language.

Cerf wanted to apply this analogy to networking. He thought the best solution would be a universal addressing system. The gateway system is the system that receives the data. This gateway could receive a packet and strip off the outer “envelope.” Then the gateway could apply a new outer envelope written in “language” of the new network. That way each network could understand how to deliver the data without having to change the data to match the needs of different networks.

These technical “post offices” were called “gateways.” The term gateway has always meant a device that connects networks with different technologies. To this day, routers (the devices that connect TCP/IP networks) are still called gateways.

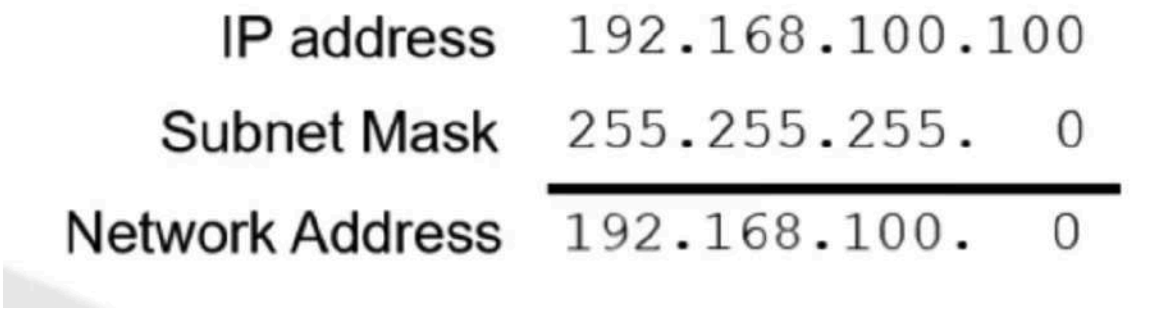
If we apply this analogy directly to TCP/IP, the local network would be like a “country.” Each network has a network address that works like a country code. Each node on the network has a node address that identifies the node. Millions, maybe billions, of nodes have the same node address. But on each network, that node address is used only once.

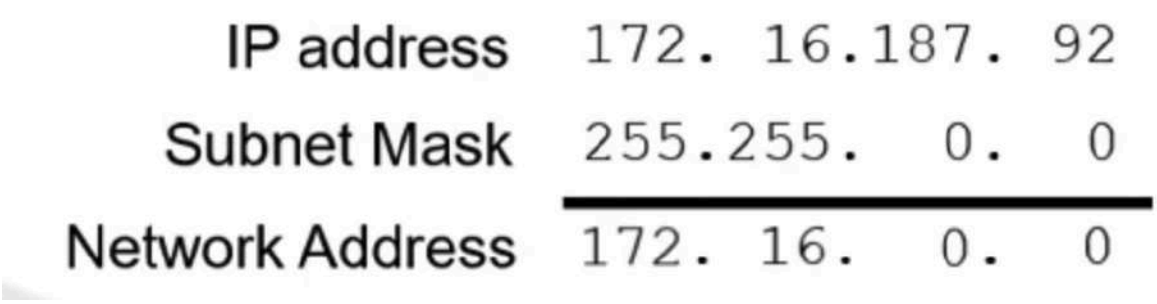
Rules of TCP/IP

Each device must have an IP address and a subnet mask. Without the subnet mask, there's no way to figure out how much of the IP address on the left is the network address. With the default subnet masks, there are no octets in the subnet mask that have both ones and zeros. Each octet is either all ones or all zeros. That means that in a default subnet mask, you will only see the numbers 255 on the left or zero on the right. And these are the three default subnet masks you will see, 255.0.0. 0, 255. 255.0.0, 255. 255. 255.0.

Any number in the IP address that's in the same position as a 255 in the subnet mask is part of the network address. Any number that's in the same position as a zero is part of the host address. The process computers used to compare the IP address to the subnet mask to find the network address is called ANDing. When you're doing ANDing with just the default subnet masks, I call that basic ANDing. Here's how

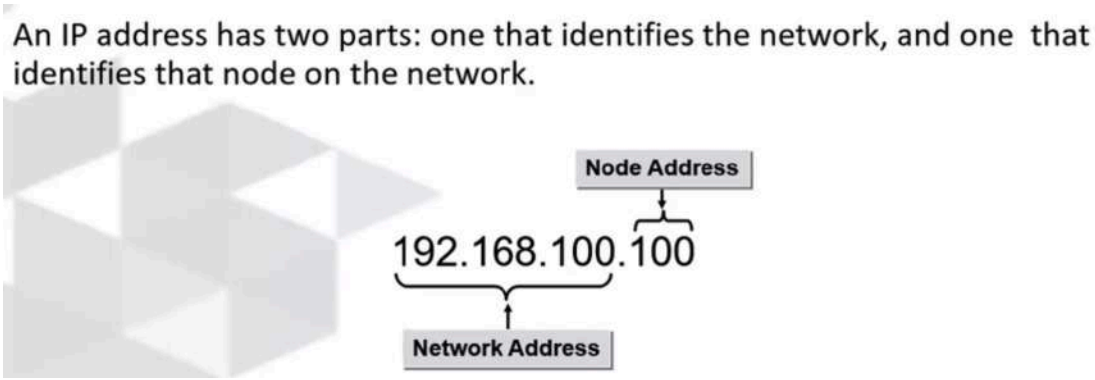
you do it. You write down the IP address. And then below that, write down the subnet mask and try to line up the dots for each octet. If the number and the subnet mask is 255, use the number on the IP address as the number for that octet and the network address, if the number in the subnet mask is a zero, uses zero as the number for that octet in the network address.



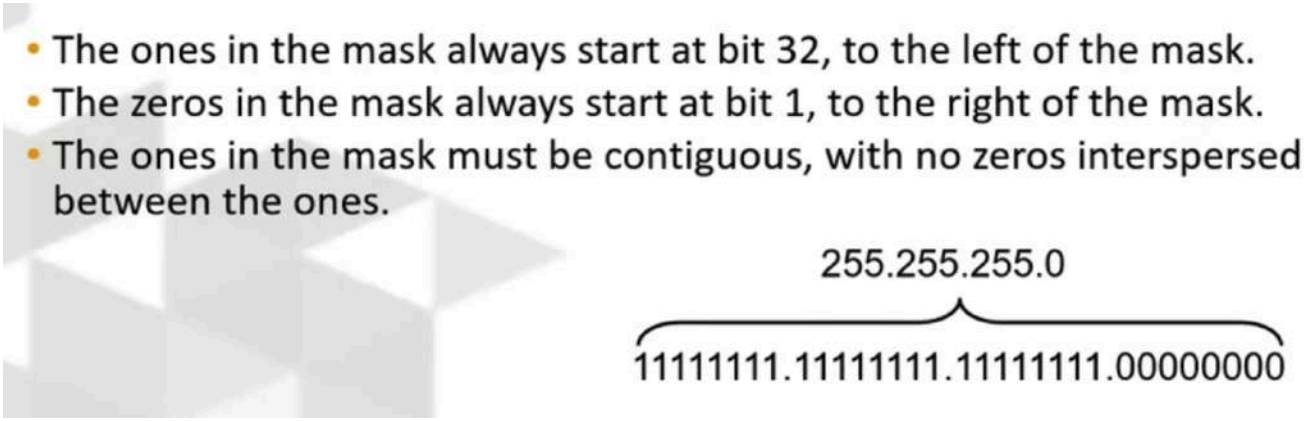


Subnet Mask

Remember an IP address has two parts, one that identifies the network called the network address, and one that identifies that node on the network called the node address. If part of the IP address is the network address, and the part that's left is the host address, how do you know which numbers go with which parts? We use the cellphone analogy to talk about IP addresses, and a cellphone it's easy. The country code is at the beginning of the number, it starts with a plus and then a 1 to 2 digit country code. With TCP IP, the network address is also at the beginning meaning on the left side of the IP address. So, it could look like this or it could look like this, the reality is there's no way to know how many of the digits in the IP address belong to the network address unless you know the subnet mask.

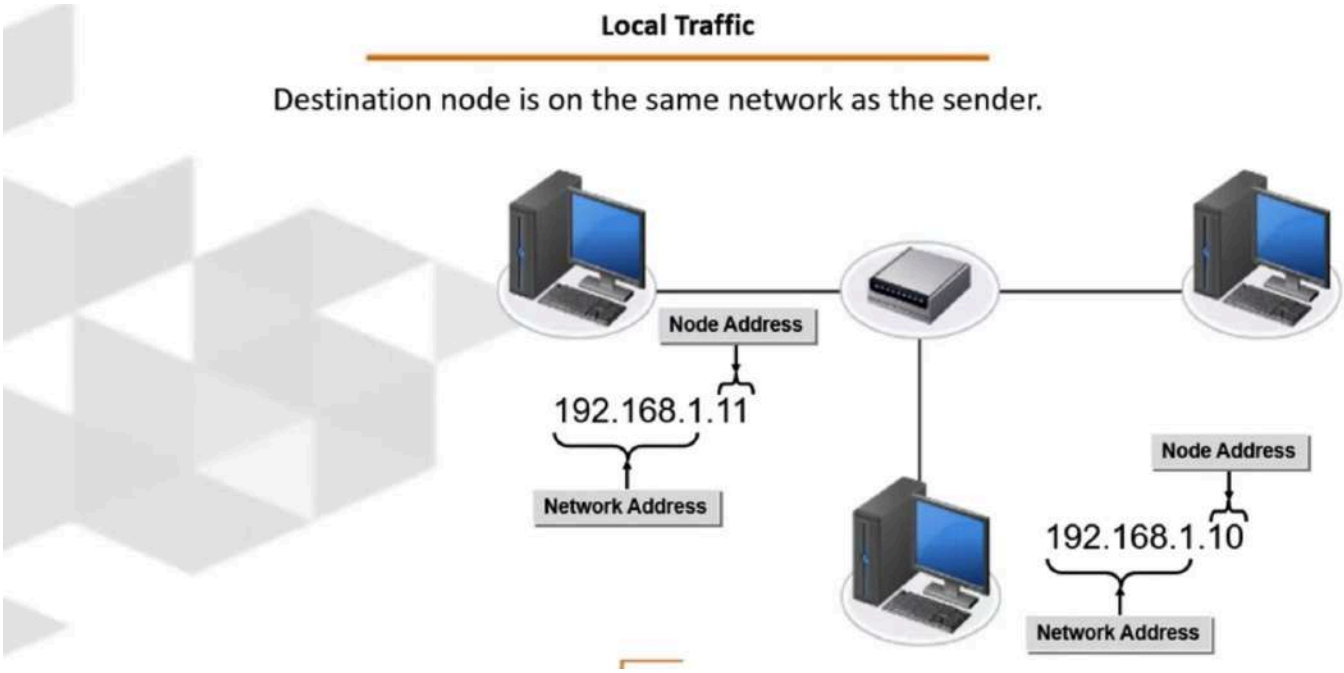


The subnet mask is a 32-bit number, also written in dotted decimal form, four octet will only see the numbers 0 through 255. The difference between the IP address and the subnet mask is that in the subnet masks, all the ones in binary are on the left side. All the zeroes in binary are on the right side, and the ones in the mask must be contiguous, meaning all in a row. So, it's all ones on the left until at some point it stops being ones and it starts being zeros, but in binary you won't see like (1, 0), (1,0) like you would in an IP address. The rule is that every bit in the IP address for which there is a one in the subnet mask is part of the network address. Every bit in the IP address for which there is a zero in the subnet mask is part of the host address.



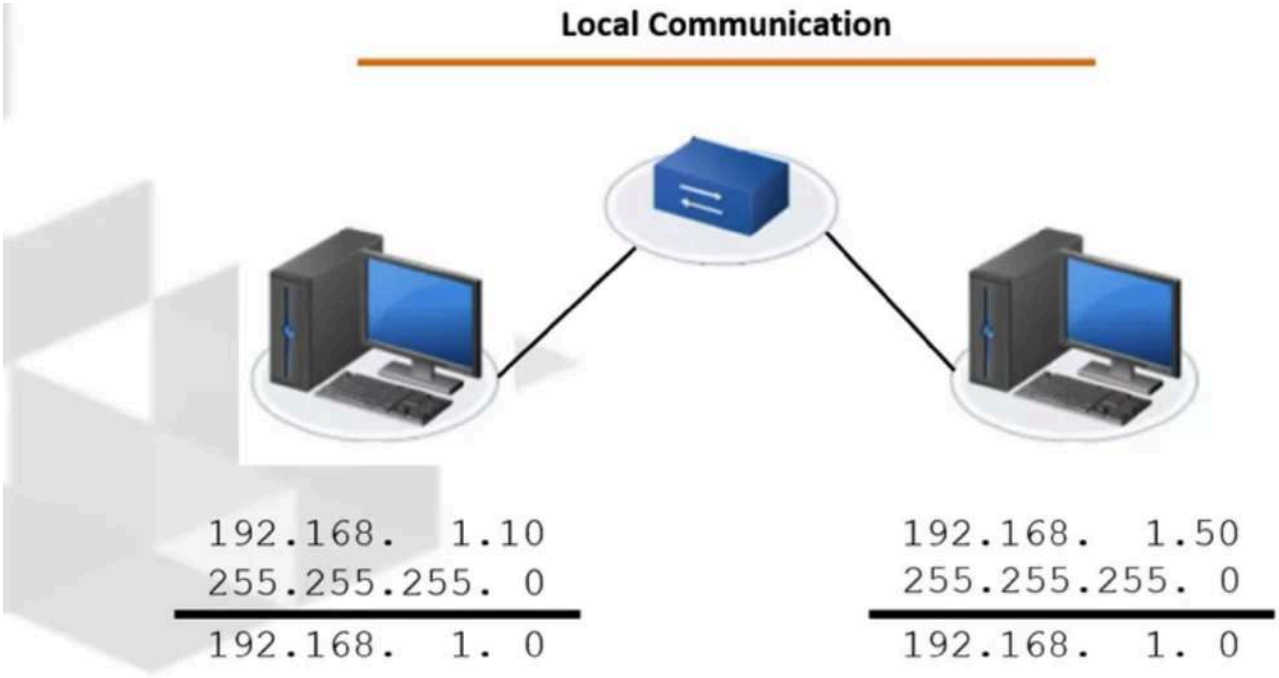
Local or Remote? Part1

When you're troubleshooting TCP/IP, the best method is to put yourself in the position of the sender, and then think through how the data is supposed to move through the network. The main thing the sender wants to know or needs to know is if the receiver is local or remote. If the receiver is local, then the data can go directly through the switch. If the receiver is remote, the data needs to be sent through the router. Let's take a look at local traffic. So with local traffic, the destination node is on the same network as the sender, that means it's local. Local fundamentally means that the sender and the receiver have the same network address. And then if that's the case, the traffic is just going to go through the switch to the destination. So how would we figure that out if we were troubleshooting a computer?



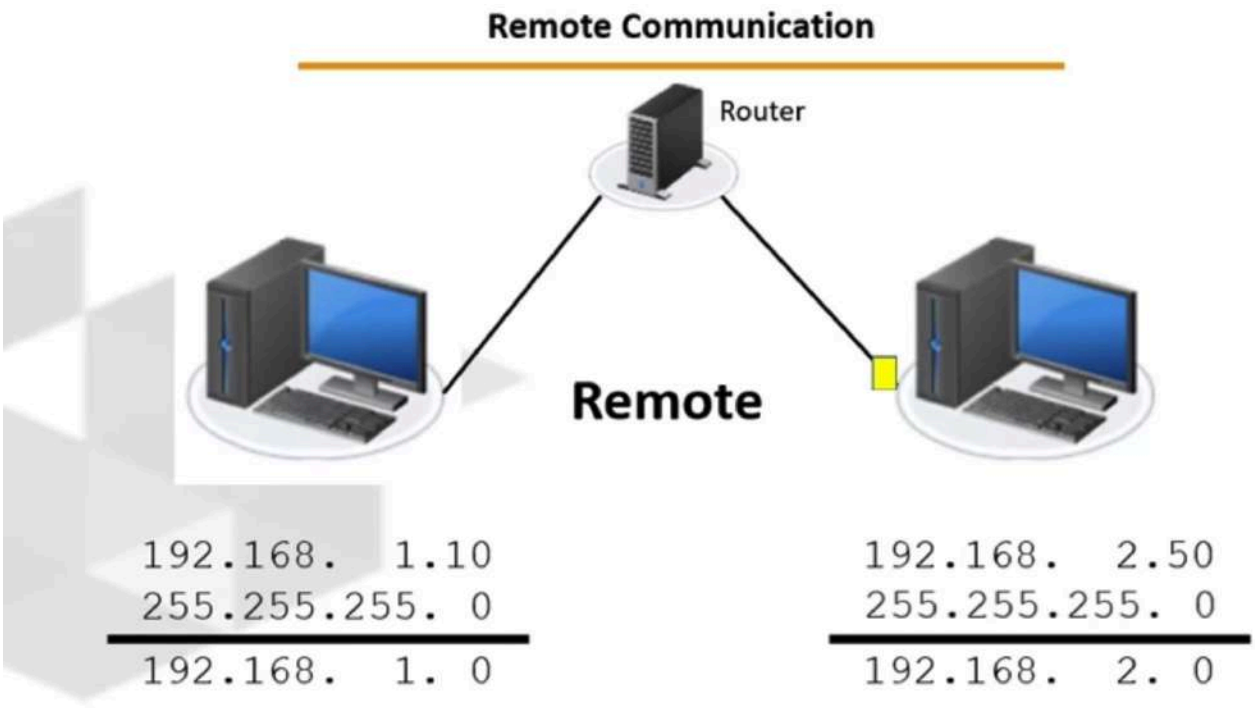
Let's take a look at an example. Suppose I'm sitting at a sender on the left and it has an IP address of 192.168.1.10 with a sub net mask of 255.255.255.0. And this computer is going to communicate with another computer that has the IP address of 192.168.1.50. The sending device first needs to figure out its own network address. Now of course, devices just know their own network ID, but if you're troubleshooting TCP/IP, you need to start by figuring out what Network ID the sending computer has using basic ending. So we would do our basic and find out that 192.168 and 1 are all part of the Network ID and come up with a Network ID or network address. Those two terms are interchangeable. 192.168.1.0, so my sending computer is on the network 192.168.1.0. But what network is the receiving computer on? Now the sending device doesn't know what subnet mask has been set up on the receiver. But if both devices are properly set up and they're on the same network, the receiver would be using the same subnet mask as the sender. That's why you always use the sender subnet mask to do your ending. And that's an important idea that you need to remember. If you're sitting at a computer troubleshooting, you will only be able to see that device's subnet mask. Most industry tests will only give the sender subnet mask. They expect you to understand that's the one to use to evaluate the receiver. So in our example, the sender is going to use basic ending to compare its subnet mask to the receiver's IP address. So it takes its subnet mask of 255.255.255.0, uses basic ending with the 192.168.1.50 address, and comes up with a Network ID of 192.168.1.0. If the network address of the sender and the receiver are identical, they're local. So the sender is going to send an ARP broadcast to find the receiver's MAC address. ARP is the Address Resolution Protocol, and it's used to

match an IP address to a MAC address. And so 192.168.1.10 will send out a broadcast. 192.168.1.50 what is your MAC address? The sender knows the receiver will get the broadcast because broadcast go out to the whole network and they're on the same network. So every computer on that network is going to get that broadcast, they're all going to pass it up to the network layer, but only the computer with IP address 192.168.1.50 is going to process that at the network layer. And it's going to reply with an ARP broadcast basically saying, hey, I'm 192.168.1.50, and my MAC address is whatever. Once the sender gets the reply, it's going to send the data to the receiver, and it goes right through the switch to the receiving computer. So, that's how this works when the two devices are local.

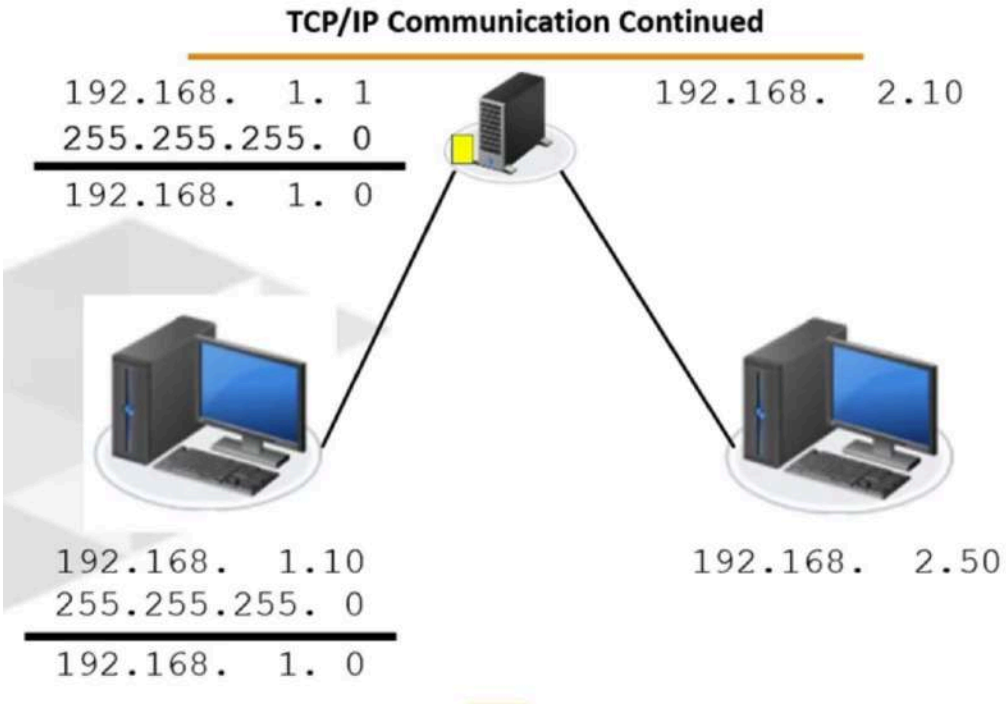


Local or Remote? Part2

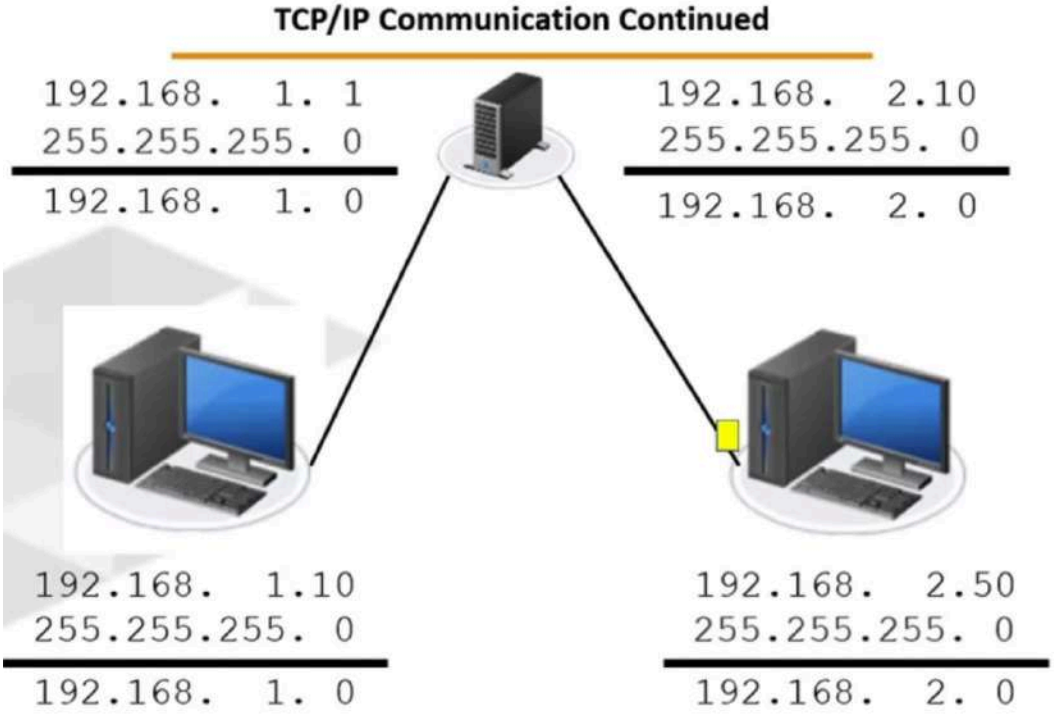
If the sender and the receiver have different network addresses, they are on different networks. Datacenter device on a different network is called remote. Let's take a look at an example. Suppose the sender has an IP address of 192.168.1.10, and a subnet mask of 255.255.255.0, and it wants to send data to 192.168.2.50. The sender does its basic ending and comes up with, hey, my network ID is 192.168.1.0. I'm trying to talk to 192.168.2.50. I use my subnet mask to evaluate the receiver and I come up with their network ID as being 192.168.2.0. Not my network ID. Unless those numbers are identical, the other device is remote, and in that case the data has to travel through a router.



Now we want to dive in a little bit more in-depth to see what happens when those two devices come up as remote. Let's dive a little deeper. Let's say that our sending computer, it still has that IP address, 192.168. 1.10, subnet mask 255.255.255.0. But it has a default gateway of 192.168.1.1, which is the address of the router and it wants to communicate with 192.168.2.50. Of course it starts out with, what's my network address? My network address is 192.168.1.0. Well, how about the receiver? What's the receiver's network address? I use my subnet mask to figure that out and I come up with 192.168.2.0. Not my network. It's remote. At that point, it's going to have to send the information to the router, which is its default gateway. To send data to a remote network, the sender must have a default gateway configured. Luckily, our sender does. If the sender has a default gateway, then it does the same check for the default gateway. Remember, routers are not exempt from any of the rules of TCP/IP. Our sender is going to use its own subnet mask to evaluate the default gateway, and it comes up with network ID 192.168.1.0. Hey, that's my network address. I'm local to the router. I can send this information to the router. At that point it's going to do an up broadcast for the router's MAC address. Hey, 192.168.1.1. What's your MAC address? The router is going to reply, hey, my MAC address is this. Then the data can be sent to the router. Now the router gets that data in



In our example here, the router has another network card with an IP address of 192.168.2.10. It applies it's subnet mask to that network card, and it says, oh, wow, that network card is on network 192.168.2.0. I've got to get this data to 192.168.2.50. I wonder what network that's on. Remember, at this point now the router has become the sender. It's going to use its own subnet mask to evaluate the receiver, and it comes up with, the receiver is on 192.168.2.0. Hey, that's my network. The router is going to do an art broadcast, 192.168.2.50, what's your MAC address? 2.50 is going to reply. My MAC address is blah, blah, blah. Then the packet comes to the receiver.

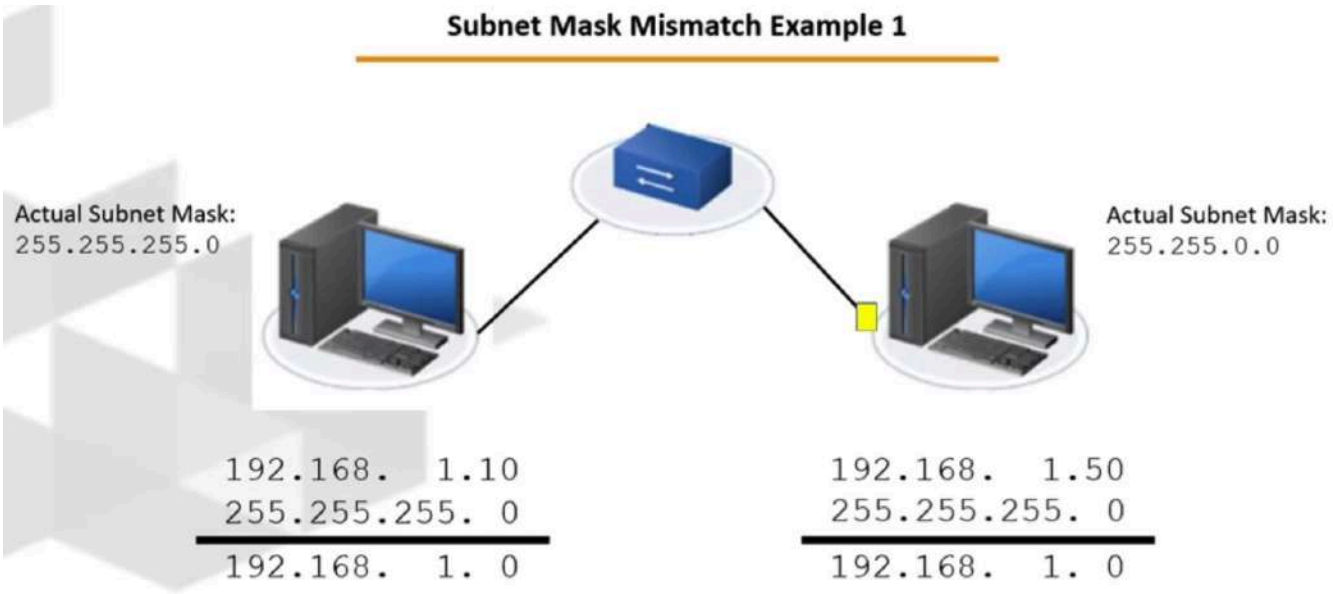


Now, watch out. When two IP addresses are remote. Always check the network address of the router. I can't tell you how many people I see that don't do that. Don't assume the router is correctly configured either on the client or on the router side. If something isn't working, it means there's a problem. Maybe that's the problem. Anytime a host can't send data out of the network and the router isn't physically powered off or broken, I would suspect that the router looks remote to the client. If the client does the basic ending and the router looks remote, the client can't send the data at all and you're done.

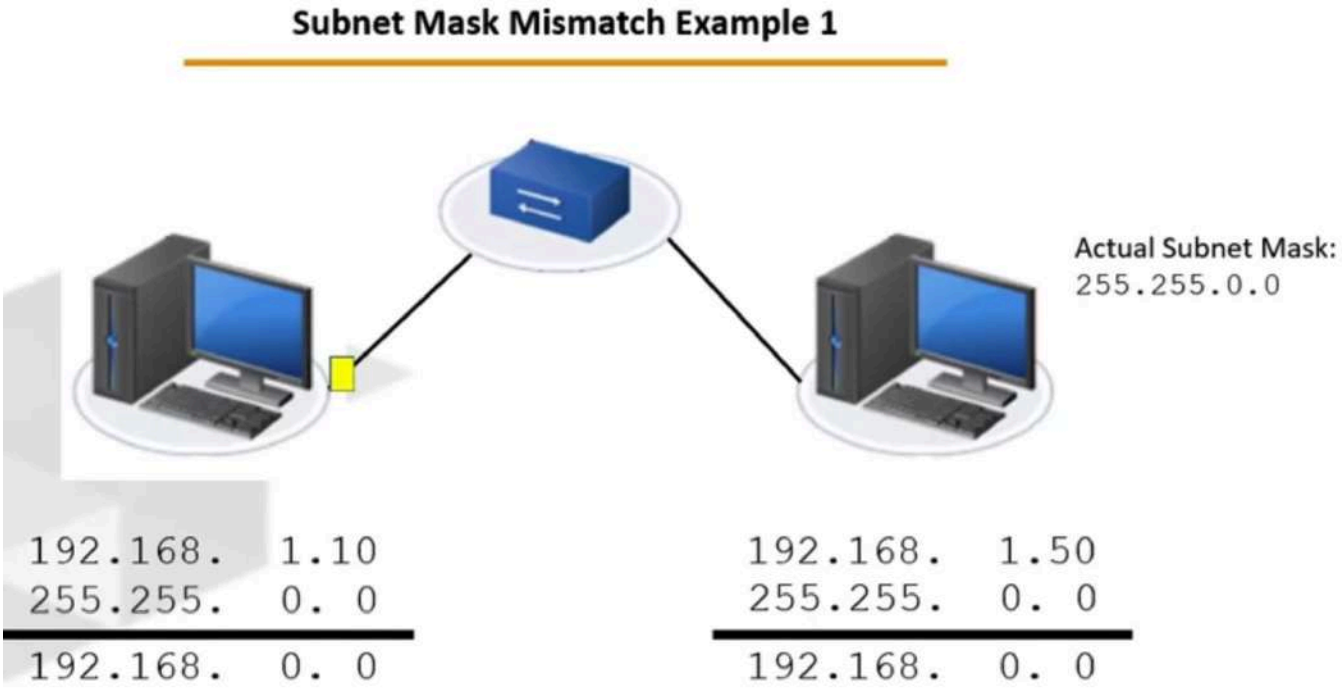
Local or Remote? Part3

nd we're going to take a look at what happens if two clients on the same network use different subnet masks. So how do we know the receiver is using the same subnet mask as the sender and what happens if it's not? We don't know if the receiver is using the same subnet mask as the sender. We assume if everything is set up properly they should be. But what happens if it's not? The answer to that depends on the IP addresses and the subnet masks being used. So we're going to look at a couple of examples.

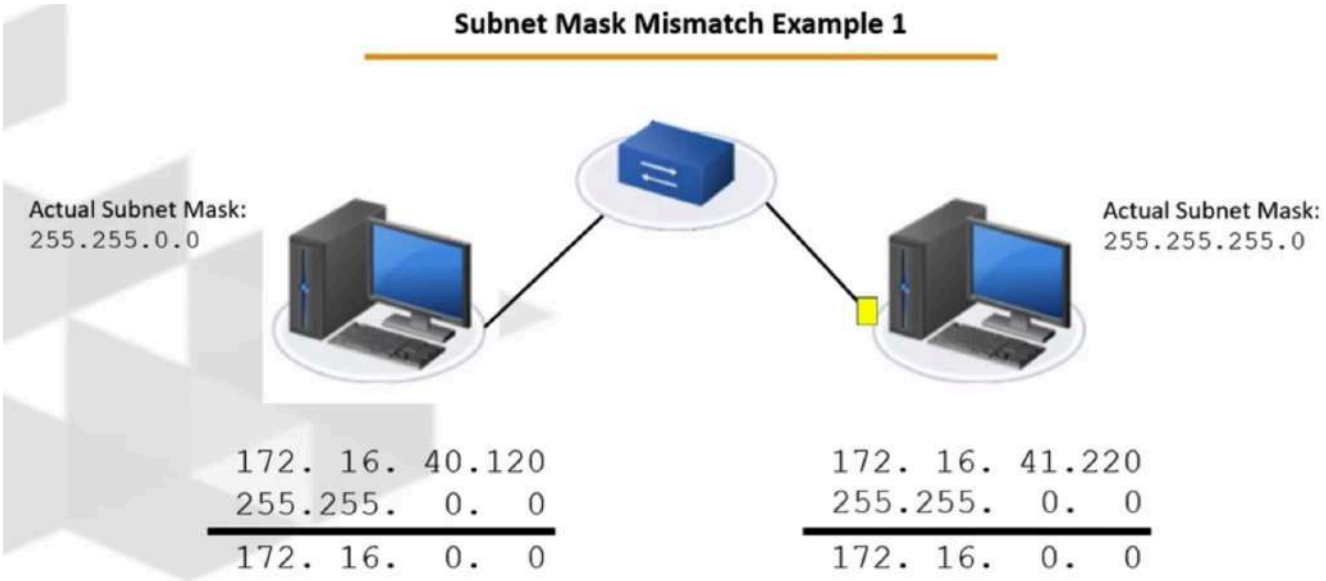
The answer to that depends on the IP addresses and the subnet masks being used. So we're going to look at a couple of examples. So for the first example, we're going to take our 192.168.1.10 computer with a subnet mask of 255.255.255.0. And it wants to communicate with 192.168.1.50 but that actually has a subnet mask of 255.255.0.0. What's going to happen? And for this example, let's assume these two computers are actually local. And what I mean by that is that they are physically connected to the same switch. So what happens when they try to communicate? The sender is going to do its basic ending like normal. So it says, all right, well what network am I on? Well, I'm on the 192.168.1.0 network. What network is the other computer on? So it uses its subnet mass to evaluate the other computer. And it says, from my perspective, looks like the other computer is on 192.168.1.0, I can just send the traffic and it works. So now that data gets to 192.168.1.50.



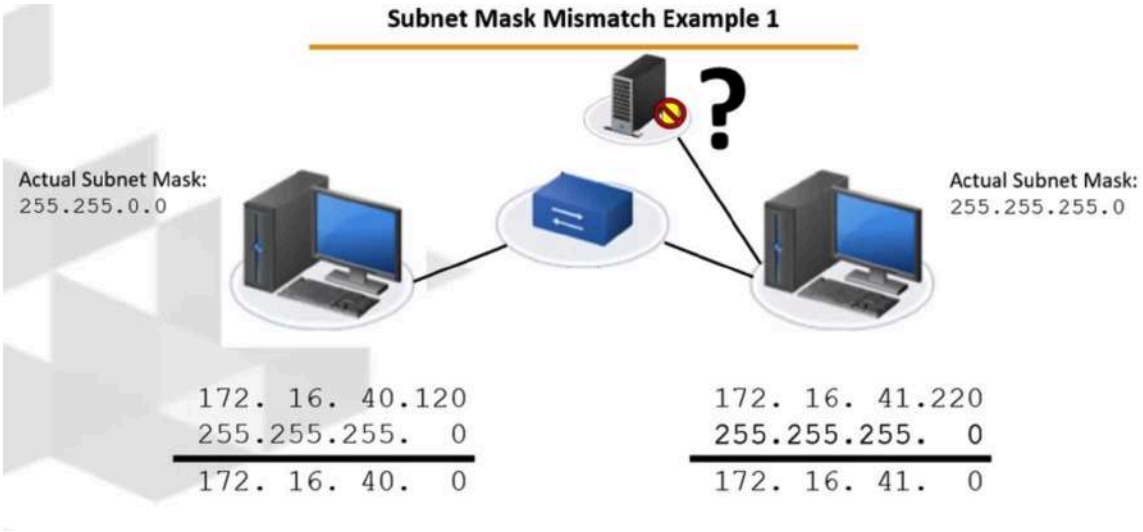
And if it's going to reply, then in that case now 192.168.1.50 is our sender and we're going to reverse the process. So 192.168.1.50 takes its subnet mask, the 255.255.0.0. So I'm on network 192.168.0.0, I wonder what network 192.168.1.10 is on? Well, let me use my subnet mass to evaluate, and look they're also on 192.168.0.0. Looks like we're on the same network, I can just send the reply back to them. And in this particular example, all the numbers work from both sides. The computers can correctly figure out that they're local and probably no one will ever even figure out that there's a problem.



So in this example let's suppose the center has an IP address of 172.16.40 .1.20 with a subnet mask of 255.255.0.0. The receiver has an IP address of 172.16.41.2 20 but their subnet mask is 255.255.255.0. And these computers are also physically local. They're connected to the same switch, what happens when they try to communicate? So the sending computer on the left is going to do its basic ending and say I'm on network 172.16.0.0, I wonder what network the other computers is on? Well, let me use my subnet mask to evaluate that computer. And it comes up with, well, that computer is also on the 172 16.0.0 network, I'm just going to send the data over.

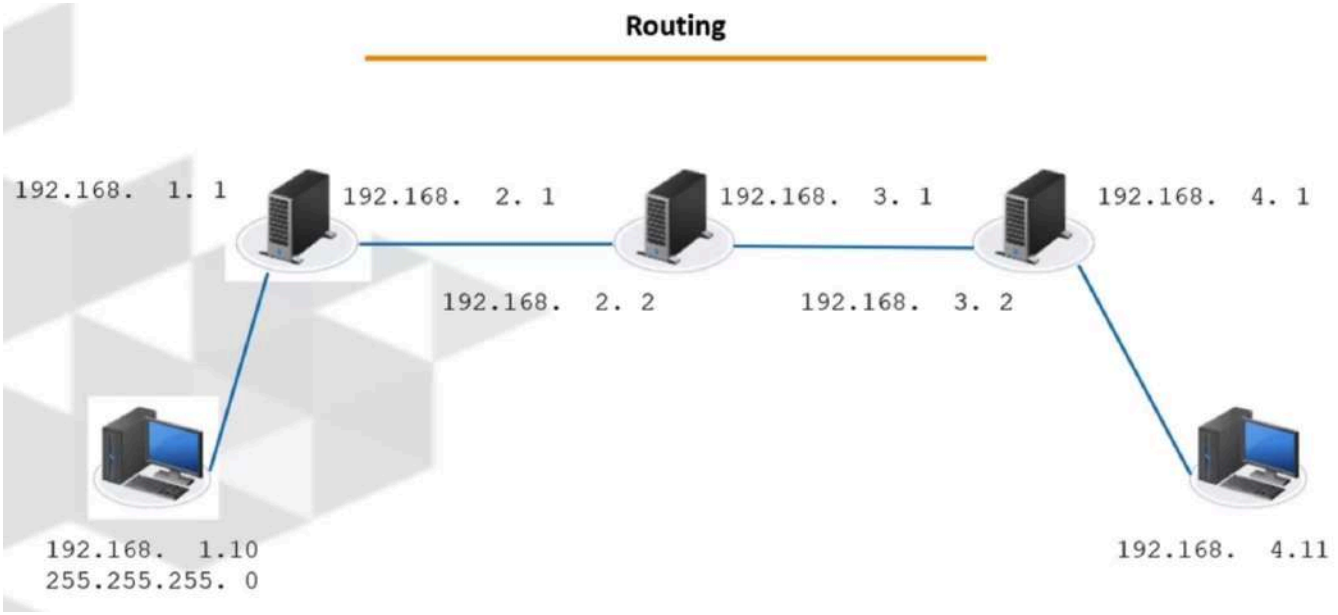


So now the other computer 172.16.41.220 gets the data and now it wants to reply. So we're going to have to do the same thing but we're going to have to do it in reverse, right? So that computer says, all right, my IP address 172.16.41.220, my subnet mask 255.255.255.0, what network am I on? I'm on the 172.16.41.0 network. How about the computer that I want to reply to? Well, let me take a look. It looks to me like they're on the 172.16.40.0 network. It's not my network, we must be remote. So I need to send the data to my default gateway. So we put a router into the picture and it's going to send that up to the router. And the default gateway is going to be like I don't know what to do with this because where you're sending it to isn't on the other side. And so the reply is never going to reach that 172 16.40.120. And that's why when you're troubleshooting TCP IP, it's very important to go slow, check each host one at a time, mentally put yourself in the position of the computer. What is happening at that point? Just because you didn't get a reply doesn't mean the data didn't reach the receiver. All it means is that either the data didn't reach the receiver or the reply didn't reach its receiver.

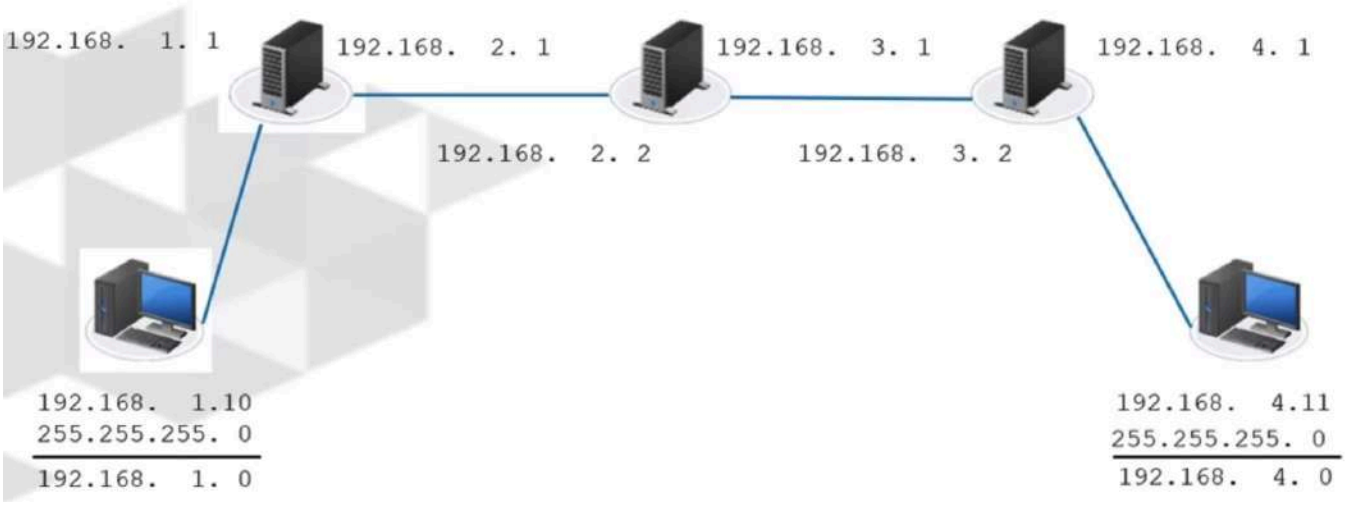


Routing

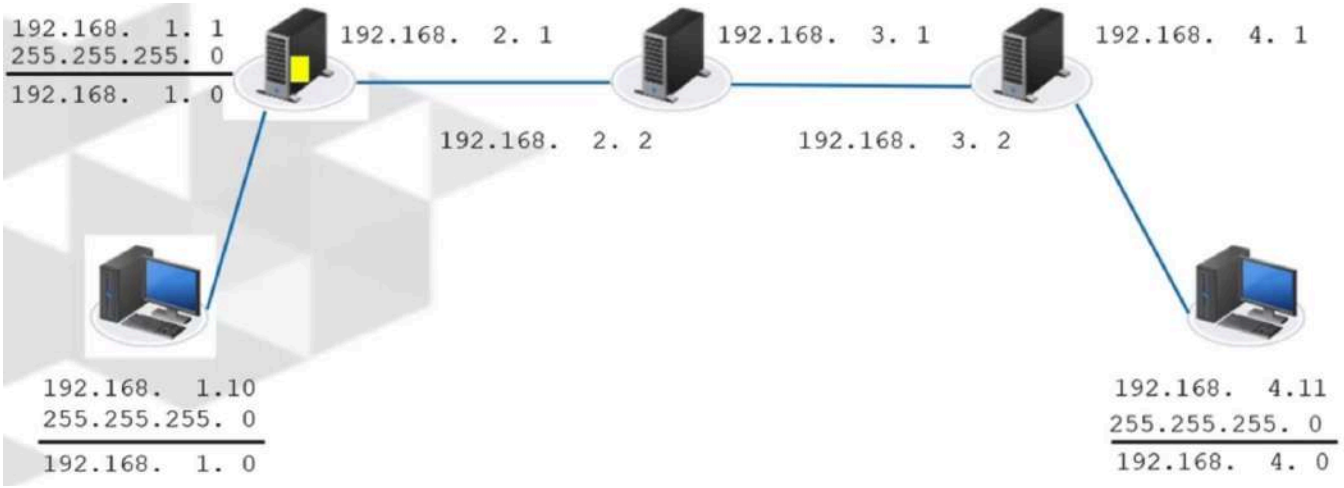
ou know how to find the network ID and you know how to answer that very important question, local or remote. Now it's time to take a look at what actually happens when there's more than one router in between the sending device and the receiving device. I've put together a very simple network for you. Our sending device is in the bottom left-hand corner with an IP address of 1921681.10, subnet mask of 255255255.0. That computer has a default gateway of 1921681.1. That first router has another network card that has an IP address of 1921682.1. It's connected to the router in the middle that has two IP addresses. One which is 1921682.2, the other is 1921683.1. That in turn is connected to another router that has two IP addresses. One of them is 1921683.2, the other is 1921684.1. Finally, we have our receiver, which has an IP address of 1921684.11.



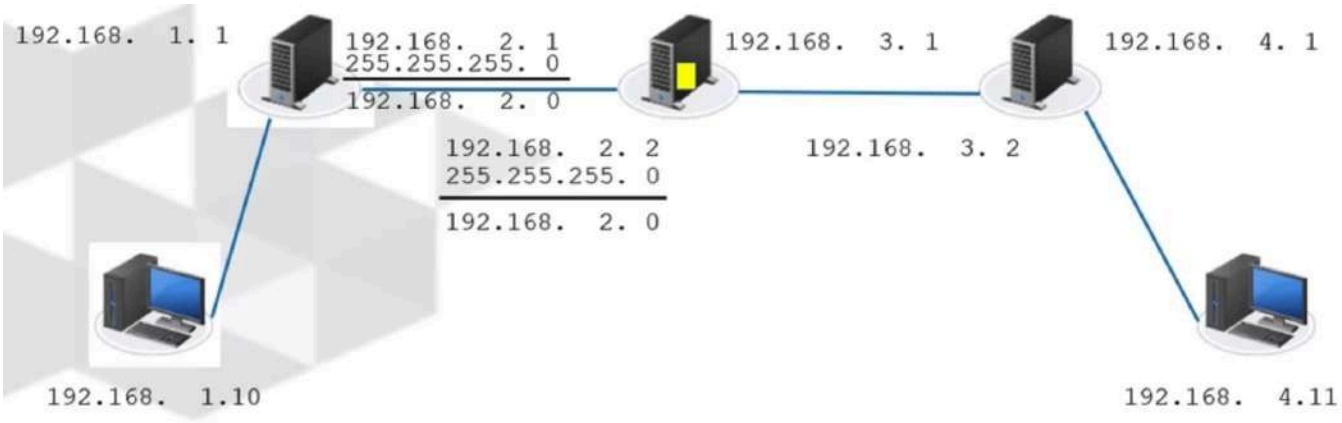
Let's take a look at what actually happens when 1921681.10 wants to send a message to 1921684.11. Of course, we start out with our sender. What network I'm I on? It's going to do its basic ending and say, hey, I'm on network 1921681.0. Gee, I wonder what network 1921684.11 is on. Remember it uses its own subnet mask to do the ending, and it comes up with, well, hey, that computer is on 1921684.0, not my network. I'm going to have to send it to my default gateway.



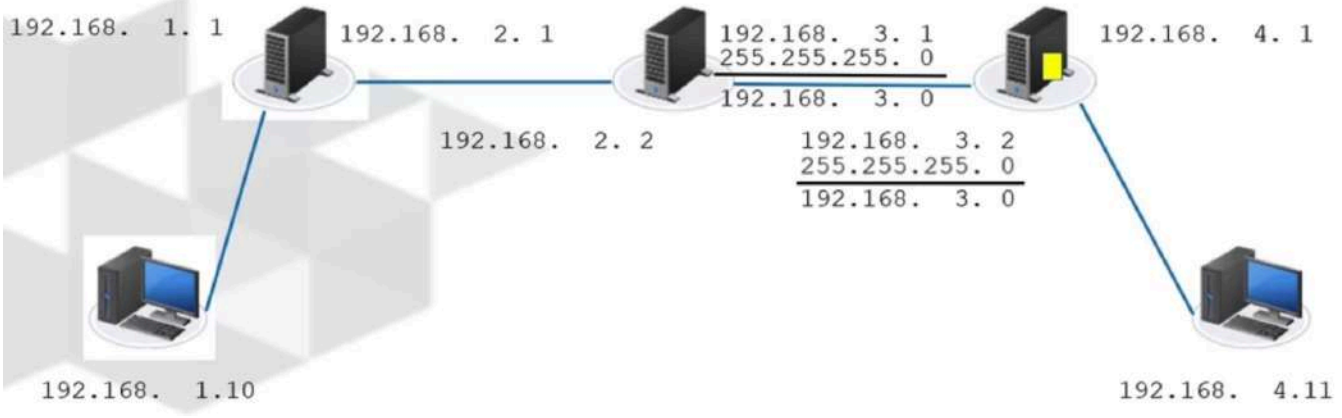
let me check the default gateway. It uses its subnet mask to do the ending for the default gateway. It comes up with the default gateway is on 1921681.0 network. Oh great, that's my network we can talk. It'll do it for the Mac address of the default gateway, and then it sends the data to the default gateway.



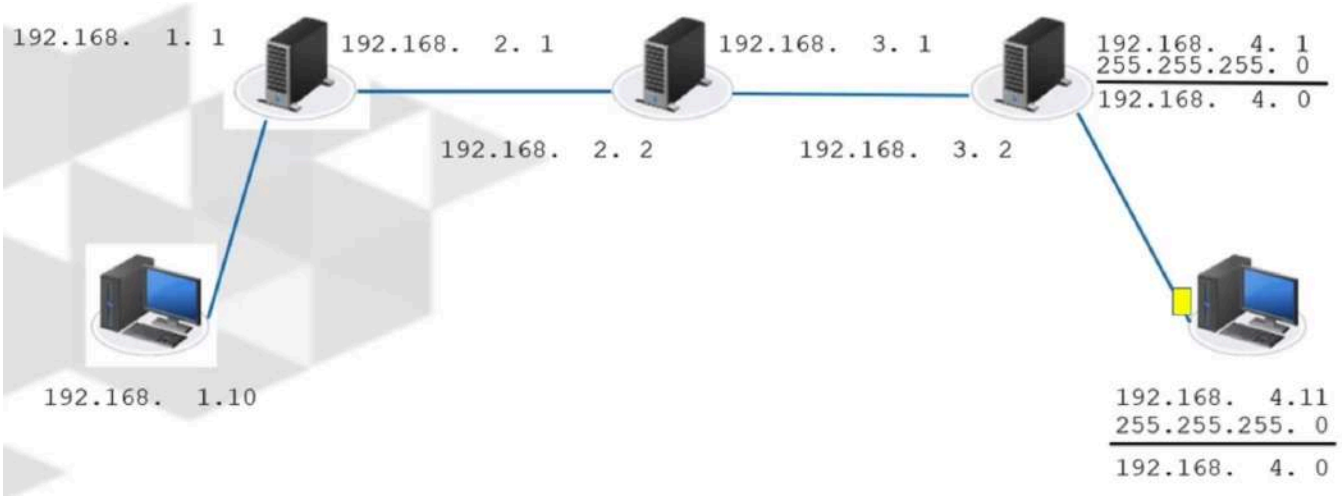
Now that first router gets it and it says, well, gee, I'm going to look at my routing table. My routing table says I should send it to that router in the middle. I have a network card 1921682.1 that happens to have a subnet mask of 255255255.0. What network am I on? Well, I'm on the 1921682.0 network. What network is the other router on? Let me use my subnet mask to check it out and come to find out, hey, that router is also on 1921682.0. Let me send it over to that router. Maybe they can get it where it needs to go.



The middle router gets it. It looks up in its routing table. The routing table says, well, you need to send it over to that router on the end. Our router in the middle says, well, gee, I've got an IP address of 1921683.1. My subnet mask is 255255255.0. What network am I on? I'm on the 1921683.0 network. Well, let me look at this 3.2 address I'm supposed to send it to use my subnet mask to do the ending. They're also on the 1921683.0 network. I can send this data over to that router. It arrives at that router.



That router looks at the destination address 19216849.11. Look, it has a network card with an address 1921684.1, subnet mask of 255255255.0. What network am I on? I'm on the 1921684.0 network. Well, how about this destination address? What network is that on? Oh, look, that's also on the 1921684.0 network. I can deliver the packet to the receiver.



That's exactly how the Internet works. The only difference is there may be thousands or millions of miles between the routers. The connections aren't made through switches, but through direct connections by the telecommunications companies. But the basic rules of TCPIP remained the same.

The only difference is there may be thousands or millions of miles between the routers. The connections aren't made through switches, but through direct connections by the telecommunications companies. But the basic rules of TCPIP remained the same. Each router only communicates with other routers that are local, meaning they have the same network address as one of that routers next. If you've always wondered, how does routing work? How does the Internet work? That's how it works.

This is like the Pony Express electronically or a baton relay race. It's really incredible how such simple rules enable communication over the entire world. In this video, we looked at routing. We followed a packet from one sender to a remote receiver with multiple routers in-between, and we saw exactly how any packet that traverses a network with multiple routers, even the Internet, really works. It all comes back to those three rules of TCPIP, and particularly that third one.

Binary Numbers

Number System

Number Systems

- Every number system has a base number:
  - Decimal = 10
  - Binary = 2
  - Hexadecimal = 16
- Each place in the number system stands for an exponent of the base.
  - Starting on the far right with the base^0 and increasing the exponent by 1 every place to the left.
  - Any number^0 is = 1
- Each position can only have one digit.
- The allowed numbers are from 0 to the base-1.

Decimal Numbers

Decimal Number System

Base	10^3	10^2	10^1	10^0
Value	1000	100	10	1

4189

$(4 * 1000) + (1 * 100) + (8 * 10) + (9 * 1) = 4189$

$(4 * 10^3) + (1 * 10^2) + (8 * 10^1) + (9 * 10^0) = 4189$

10^3	10^2	10^1	10^0
1000	100	10	1
4	1	8	9

Subnetting

The network address is an address where the node address bits are all 0s.

IP address	11000000.10101000.00000001.00001010	192.168. 1.10
Subnet Mask	11111111.11111111.11111111.00000000	255.255.255. 0
Network Address	11000000.10101000.00000001.00000000	192.168. 1. 0

The network address is always below all the usable IP addresses. The first usable IP address is always the network address +1. In this case, with a network ID of 192.168.1.0 the first usable IP address would be 192.168.1.1.

The broadcast address is an address where the node address bits are all 1s.

IP address	11000000.10101000.00000001.00001010	192.168. 1. 10
Subnet Mask	11111111.11111111.11111111.00000000	255.255.255. 0
Broadcast Address	11000000.10101000.00000001.11111111	192.168. 1.255

The broadcast address is always the last "IP address" in the network but you cannot use the broadcast address as an IP address for clients.

- The first address on the network is the network address.
- The first usable IP address is the network address +1.
- The last address on the network is the broadcast address.

Typical Class C Network

Network Address: 192.168.1.0
Subnet Mask: 255.255.255.0
Broadcast Address: 192.168.1.255

Usable Client IPs: 192.168.1.1 – 192.168.1.254

254 Client Addresses on 1 Network

Subnetting

But what if we need two networks?

We must subnet the network.

When you subnet a network, the base network address must remain.

The new subnets come from the host bits.

To make the new networks, change the subnet mask.

The network address is an address where the node address bits are all 0s.

New Subnet Addresses:

First Bit	Host Bits	Address	Subnet Address
0	0000000	0	192.168.1.0
1	0000000	128	192.168.1.128

The broadcast address is an address where the node address bits are all 1s.

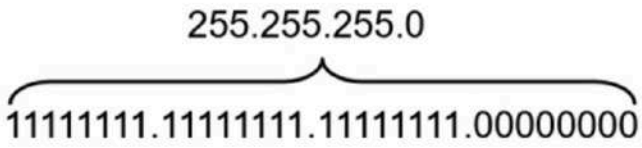
New Subnet Addresses:

First Bit	Host Bits	Address	Broadcast Address
0	1111111	127	192.168.1.127
1	1111111	255	192.168.1.255

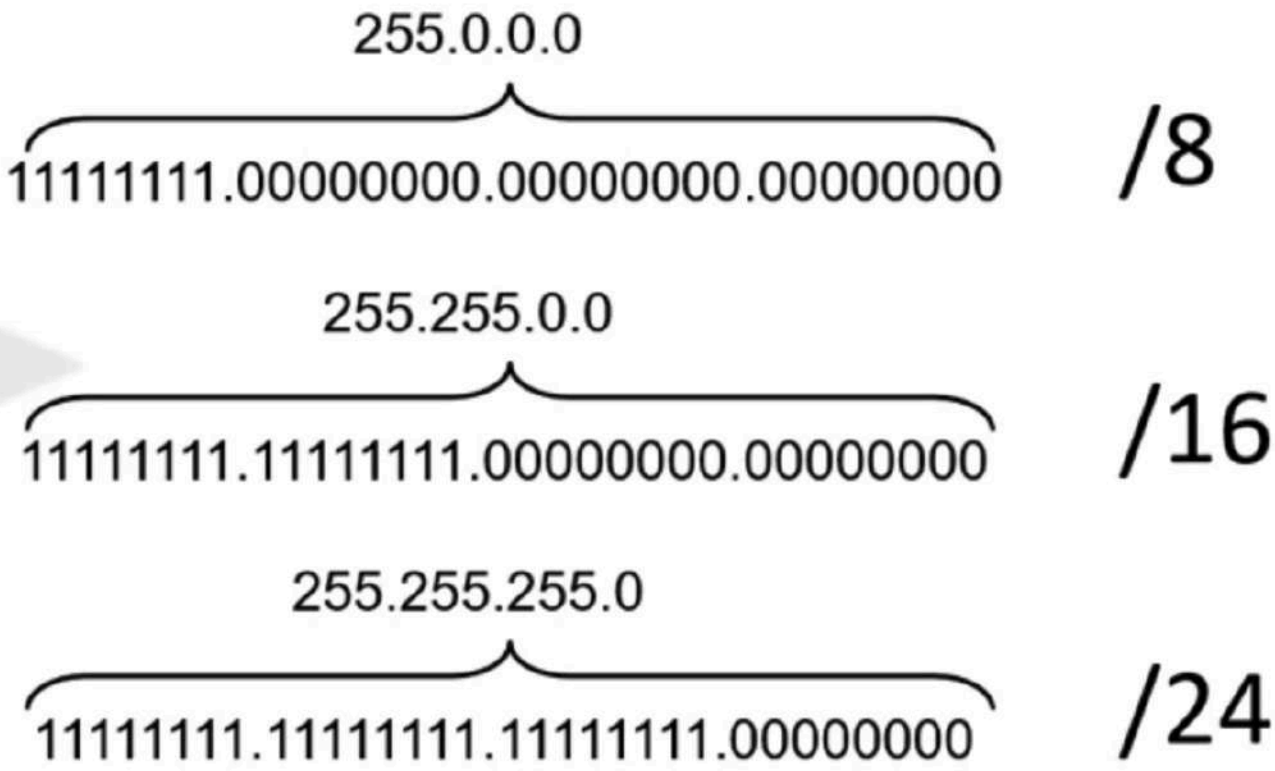
CIDR Notation

All CIDR notation is is a way of expressing the subnet mask as a slash number. We know that a binary number with eight ones is 255 in decimal. We know what the subnet mask, the ones are on the left, the zeros are on the right. We don't mix them up. At some point it stops being ones, it starts being zero.

- The ones in the mask always start at bit 32, to the left of the mask.
- The zeros in the mask always start at bit 1, to the right of the mask.
- The ones in the mask must be contiguous, with no zeros interspersed between the ones.



It's not really rocket science or difficult Math to figure out that if I have a subnet mask of 255.0.0.0, then altogether in that subnet mask, I have a total of eight ones, and of course they're going to be on the left-hand side because that's where the ones are. Somebody said, why can't we just call that /8 and everybody will know what we're talking about. You can imagine if we have a subnet mask of 255255.0.0, that's going to be /16. If we have 255.255.255.0, that's going to be /24. The CIDR notation just allows us to express the subnet mask as a slash followed by the total number of ones in the entire subnet mask.

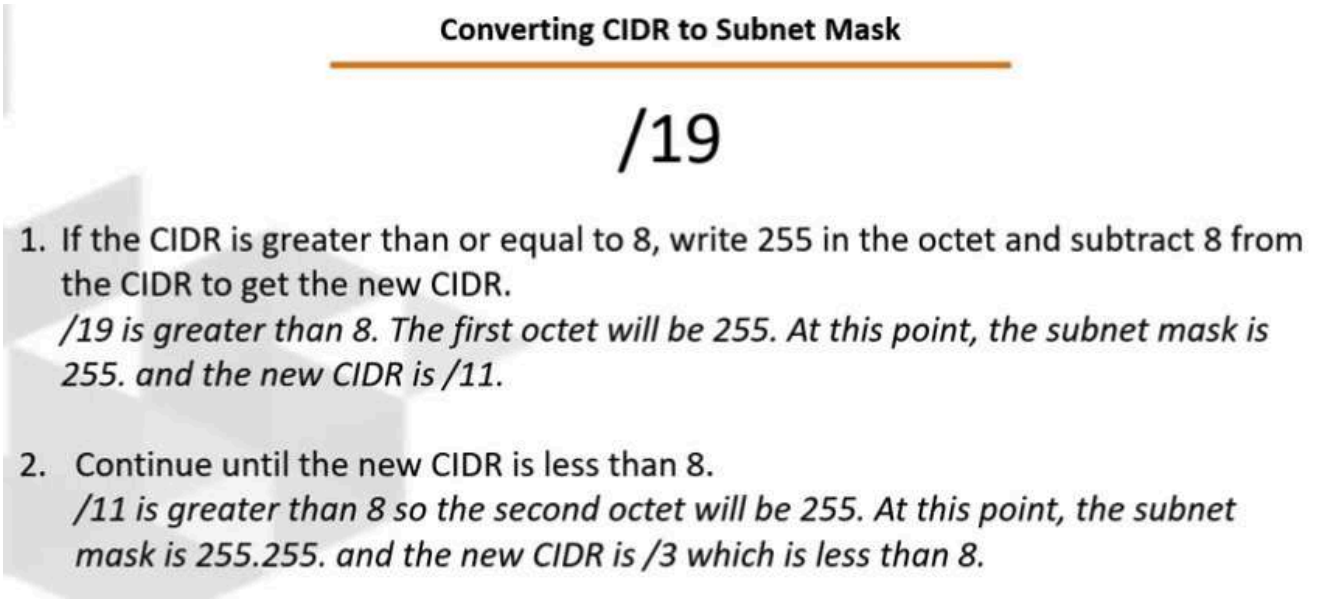


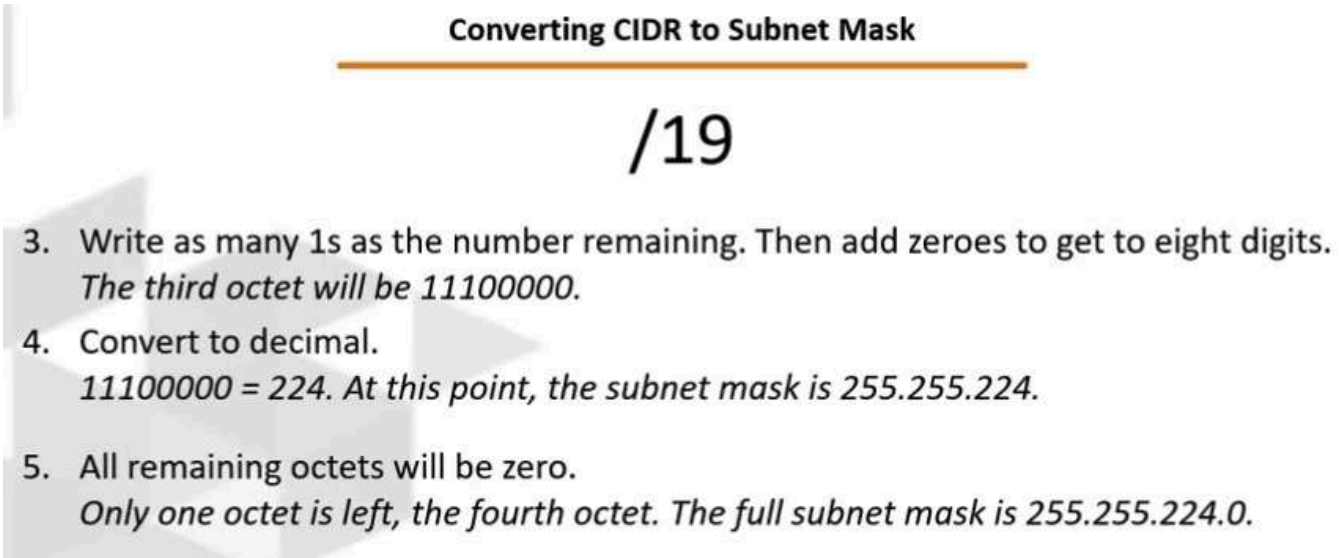
Now you might see a CIDR notation for a subnetted network. That means that that number is not /8, /16 or /24. How can we convert that to a subnet mask. E.g. what about a /19? Well, we could just write it out. You can just write out 19 ones and all the zeros and then convert each of those octets from binary to decimal. But that's a lot of writing of ones. I have a simpler way to do it. You just follow some few steps. It's a lot easier than doing all of this.

1. If the CIDR is greater than or equal to 8, write 255 in the octet and subtract 8 from the CIDR to get the new CIDR.
2. Continue until the new CIDR is less than 8.
3. Write as many 1s as the number remaining. Then add zeroes to get to eight digits.
4. Convert to decimal.
5. All remaining octets will be zero.

If your CIDR is greater than or equal to eight, you write 255 in the octet, and then you subtract eight from the CIDR to get the new CIDR and you keep doing that until it's less than eight. You write out as many ones as you have the number remaining, you add zeros to get to eight digits, you convert that to decimal, and then all the remaining octets will be zero. Remember there's only one point in a subnet mask where it stops being ones and it starts being zeros. There's only going to be one weird octet that's not 255 and not zero. But let's take a look at it. This system makes a lot more sense when you go through the example. Let's do

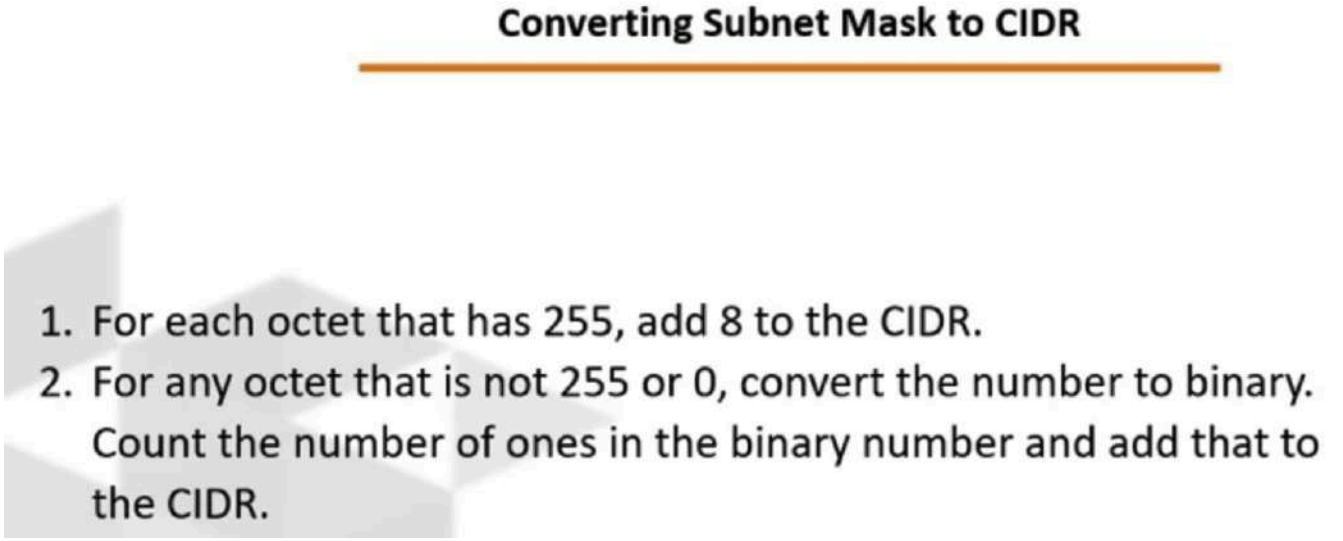
that /19. Rule Number 1 or step Number 1 says, well, if your CIDR is greater than or equal to eight, you write 255 in the octet and you subtract eight. Nineteen is greater than eight, so our first octet is going to be 255 and then our new CIDR is going to be 11. Second step says, well, we're going to keep doing that until we get to less than eight. /11 is greater than eight. My second octet is going to be 255. Right now my subnet mask is 255.255 and my new CIDR is /3 which is less than eight. I can move on to Step 3. We write down as many ones as the number remaining and add zeros to get to eight digits. I would write down three ones, I add five zeros. That gets me to eight digits. I convert that to decimal and I get it to 24. At this point the subnet mask is 255.255.224, and then anything left is going to be zero. Well, we only have one octet left, the fourth octet. My subnet mask is 255.255.224.0.

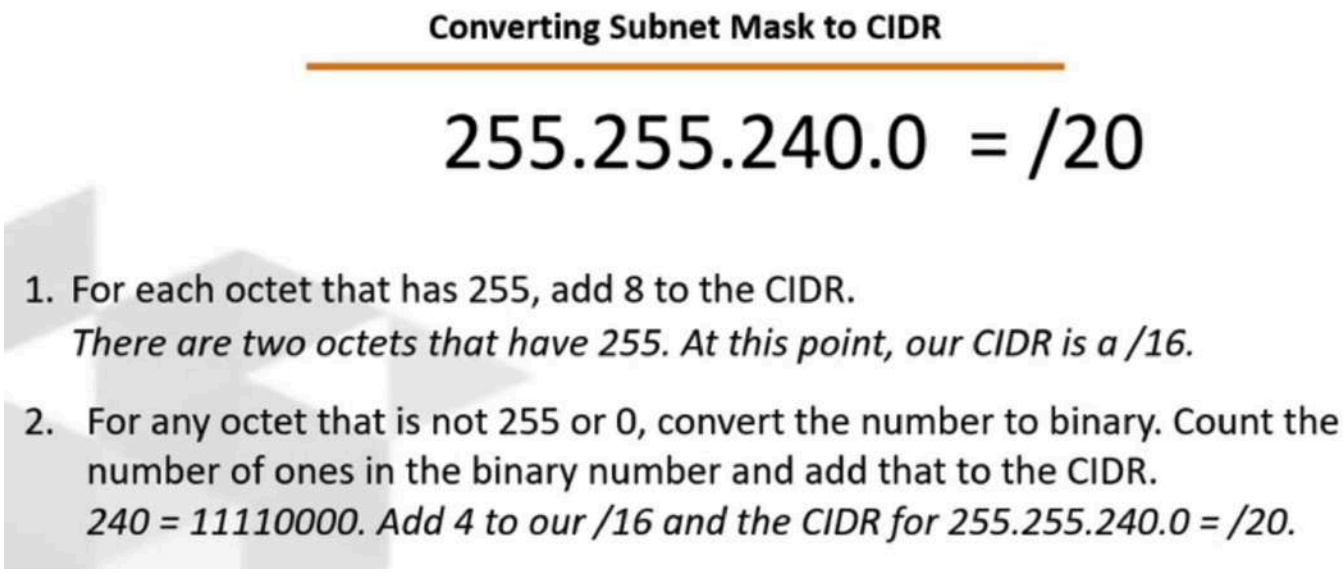




It's very rare that you would have to take a subnet mask and convert that to a CIDR. But that's also pretty easy. Just two steps. For each octet that has a 255, you add eight to the CIDR. For any octet that's not 255 or zero, convert that to binary, count the number of ones and add that to the CIDR and there's your CIDR. Let's try one. Let's suppose we had 255.255.240.0. Well, we have two octets with 255. For each of those we'd add eight to our CIDR. That gives us a /16, but now we have to figure out what's going on with that

240. For any octet that's not 255 or zero, we're going to convert that to binary. Well, that comes out to this number here, 11110000. We count up the number of ones. There are four. We add four to our /16 and we come to find out that the CIDR for 255.255.240.0 is a /20





What is the subnet mask being represented by a /23 CIDR?

1. If the CIDR is greater than or equal to 8, write 255 in the octet and subtract 8 from the CIDR to get the new CIDR.
   /23 is greater than 8. The first octet will be 255. At this point, the subnet mask is 255. and the new CIDR is /15.
1. Continue until the new CIDR is less than 8.
   /15 is greater than 8 so the second octet will be 255. At this point, the subnet mask is 255.255. and the new CIDR is /7 which is less than 8.
1. Write as many 1s as the number remaining. Then add zeroes to get to eight digits.
   The third octet will be 11111110.

1. Convert to decimal.
   11111110is equivalent to 254. At this point, the subnet mask is 255.255.254.
1. All remaining octets will be zero.
   Only one octet is left, the fourth octet. The full subnet mask is 255.255.254.0.

Another way to do-

To find the subnet mask represented by a /23 CIDR (Classless Inter-Domain Routing) notation, you need to understand that the CIDR notation indicates the number of bits that are used for the network portion of the address.
In CIDR notation, the /23 means that the first 23 bits of the address are used for network identification, leaving the remaining bits for host identification within that network.
A subnet mask is a 32-bit number where the leftmost consecutive bits are set to 1 to represent the network portion, and the rightmost bits are set to 0 to represent the host portion.
For a /23 CIDR:

- The first 23 bits are used for the network portion.
- The remaining 9 bits are used for the host portion.

So, the subnet mask can be represented as:
11111111.11111111.11111110.00000000
11111111.11111111.11111110.00000000
In decimal format, this would be:
255.255.254.0
255.255.254.0
Therefore, the subnet mask for a /23 CIDR is 255.255.254.0.

Internet Protocols

IPV4 Addresses

Background

TCP/IP was created to provide internet working for one network: ARPANET. The creators never envisioned a network as large as the Internet.
In 1993 when the Internet Engineering Task Force created the IPv4 standard, the Internet was already running out of IP addresses. At that time, every computer that connected to the Internet was on the Internet with a valid public address. A “public IP address” is an address that is directly on the Internet. At that time, companies would purchase networks from an organization called InterNIC run by the Stanford Research Institute as a registered service of the US Department of Commerce. (In 1997, that

responsibility was transferred to a non-profit organization created to manage IP addresses and DNS called American Registry for Assigned Numbers (ARIN).) Companies would purchase an entire network. If they needed more networks, they could subnet the network they purchased. But all of the IP addresses in use had to be valid public IP addresses.
Because of this, every IP address in use had to be unique on the Internet. With the numbers of Internet users increasing, even companies who bought very large networks were running out of IP addresses to assign.

IPv4 Classes

TCP/IP addresses were originally divided up into classes based on the very first few bits in the IP address. Each class was assigned a default subnet mask (if appropriate) and those were the only subnet masks recognized by Internet routers.
Here are the IPv4 classes:

Class	IP Starts With	1st Octet Decimal	Default Subnet Mask	# of Hosts	Purpose
A	0 (00000000 – 01111111)	1 – 126*	255.0.0.0	16,777,214	Large networks
B	10 (10000000 – 10111111)	128 – 191	255.255.0.0	65,534	Medium Networks
C	110 (11000000 – 11011111)	192 – 223	255.255.255.0	254	Small networks
D	1110 (11100000 – 11101111)	224 – 239	NA	NA	Multicast
E	11110 (11110000 – 11110111)	240 – 247	NA	NA	Reserved for future experiments

* The first octet can’t be 0 and the 127 network was reserved for testing TCP/IP and never used.
Every network administrator should be able to look at an IPv4 host address and know if it is class A, B or C.
Restricted Addresses
Some IP addresses have special uses and cannot be assigned to networks and hosts. They are as follows:

1. The class A network 127.0.0.0 is used for testing purposes. The most used address for testing is 127.0.0.1 which is called the “loopback address.” If you can ping the loopback address, TCP/IP is working on the device. Traffic sent to any 127.0.0.0 address is routed back to the local device.
2. The network address cannot be all zeroes. When the network address is set to 0, TCP/IP interprets the IP address as a “local” address, meaning that the data packet does not need to be transmitted through a router. For example, 0.0.0.22 identifies host 22 on the local network.
3. The host address cannot be all zeroes. The address where the host portion is all zeroes identifies the network address.

4. The host address cannot be all ones. The address where the host portion is all ones identifies the broadcast address. This address is used when nodes want to contact all hosts on the network.

Private IP Addresses

With the Internet running out of IP addresses, there was a push to solve the problem. The permanent solution to this issue will be IPv6. IPv6 uses 128-bit addresses. The address space is so large, every device on the planet could have a public address.

However, in 1993, the world wasn't ready for IPv6.

At that time, to use IPv6 every device, from the sender to the receiver, needed to support IPv6. It can take years, if not decades, to give organizations time to update all their equipment to a new standard. The problem was too urgent to wait for the world to buy new hardware and update the software. The world needed an interim solution.

That solution was private IP addresses and Network Address Translation (NAT).

In the late 1990s, the Internet Engineering Task Force directed the Internet Assigned Numbers Authority (IANA) to set aside blocks of addresses to be used in private networks.

Previously, private networks had to use public addresses. Using an address that you hadn't purchased could create a duplicate address on the Internet which would violate the rules of TCP/IP.

Even if you were able to find an address that wasn't in use, using an address that didn't belong to you could create other problems. Sooner or later, you would try to contact a resource on the Internet with an IP address that would look local to your address. The data would be directed to the local network instead of the default gateway.

To avoid this problem, the private IP addresses were removed from the pool of addresses for the Internet. As of this writing, the reserved addresses are:

Network	Addresses	Purpose
10.0.0.0 /8	10.0.0.0 – 10.255.255.255	Private Use
172.16.0.0 /12	172.16.0.0 – 172.31.255.255	Private Use
192.168.0.0 /16	192.168.0.0 – 192.168.255.255	Private Use
169.254.0.0 /16	169.254.0.0 – 169.254.255.255	Automatic Private IP Addressing (APIPA)
100.64.0.0 /10	100.64.0.0 – 100.127.255.255	Carrier-grade NAT

Anyone can use an address from the first three blocks for their private networks. They will be guaranteed not to conflict with anything on the Internet.

Automatic Private IP Addressing (APIPA) was created as a solution for DHCP clients. Prior to APIPA, if the client was set to obtain an IP address from DHCP, and no DHCP server was available, the client would have an IP address of 0.0.0.0. The client could not communicate on the network at all.

To resolve this problem, modern devices configured to use a DHCP server now support APIPA. The client tries to contact the DHCP server several times (usually five.) If the server doesn't answer, it chooses an IP address from the APIPA range. The client will send a ping to that address to make sure no other client has

already chosen that address. If there is no conflict, it uses the address. This allows DHCP clients to contact each other on the local network while DHCP is being fixed. Effectively, if you see an IP address in the 169.254.0.0 /16 network, it means there is a problem with DHCP.
Carrier-grade NAT is a special type of NAT that is used in wireless networks, particularly cellular networks. For practical purposes, these are private addresses used by cellular telecommunications companies. They should not be used privately by any other entities.

Network Address Translation

Creating the private IP address blocks resolved the problem of supplying enough IP addresses for private companies.
But with possibly billions of devices all using the IP address 192.168.1.10, how can data be delivered? How do hosts on the Internet reply to an address that isn't unique?
Network Address Translation (NAT) enables that system to work.
Routers are devices that are connected to two or more different networks that can pass information between them. If data comes into one NIC on a router destined for a different NIC, the router passes the data to the destination network.
NAT routers work differently.
With NAT routers, one side of the router is a NIC connected to a private network that uses private IP addresses. The other side is (theoretically) connected to the Internet. As the private clients send data to the Internet, the NAT router repackages the data with the IP address of the NIC connected to the Internet. The data is tagged with information (usually a port number) that the NAT router uses to track which private IP should get the reply. Since the router's Internet IP is a public IP, it's unique on the Internet. Replies return to the NAT router's public IP. The NAT router uses the tag (port number) on the reply to route the data back to the right host.
Here's an exercise you can try right now.
Open a command prompt on the computer you're using to read this lesson. Execute the command **ipconfig**. Notice your IP address. Here's mine:

```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::2989:313b:9e5b:7848%19
   IPv4 Address. . . . . . . . . . . : 192.168.1.71
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1
```

Now go into your favorite search engine and search for "What's my IP." Here's what I got back:

That's NAT in action.
NAT allows hundreds, sometimes thousands, of devices to share one public IP address. As the Internet continued to run out of addresses, telecommunications companies have started to implement multiple layers of NAT.
Try this.
Open a command prompt on the computer you're using to read this lesson. Execute the command **tracert *domain***. (Replace the word "domain" with any domain name of your choosing. I'm going to use yahoo.com.)
This is what I got:

```
PS C:\Users\Shadow> tracert yahoo.com

Tracing route to yahoo.com [74.6.143.26]
over a maximum of 30 hops:

  1     4 ms     4 ms     5 ms  192.168.1.1
  2    17 ms    15 ms    10 ms  10.1.8.1
  3    15 ms    16 ms    14 ms  100.120.245.128
  4    22 ms    14 ms    15 ms  100.120.245.57
  5    18 ms    18 ms    18 ms  nyrkbprj01-ae3.0.rd.ny.cox.net [68.1.5.157]
  6    42 ms    20 ms    24 ms  68.105.31.82
  7    33 ms    50 ms    33 ms  ae-1.pat2.bfw.yahoo.com [209.191.64.165]
  8    41 ms    36 ms    30 ms  et-0-1-1.msr1.bf1.yahoo.com [74.6.227.65]
  9    30 ms    34 ms    32 ms  et-0-1-0.clr2-a-gdc.bf2.yahoo.com [74.6.122.25]
 10    26 ms    27 ms    29 ms  lo0.fab4-1-gdc.bf2.yahoo.com [74.6.123.241]
 11    26 ms    28 ms    31 ms  usw2-1-lbb.bf2.yahoo.com [74.6.98.139]
 12    32 ms    28 ms    30 ms  media-router-fp74.prod.media.vip.bf1.yahoo.com [74.6.143.26]

Trace complete.
PS C:\Users\Shadow>
```

The first "hop" (each hop is a router) is my home wireless router. Hop 2 is a private IP address of the first NAT router from my ISP. But notice that hops 3-4 are carrier-grade NAT addresses. That means my ISP has three levels of NAT between my home network and the Internet. The 70.181.51.117 that resources on the Internet see as "my" IP address is really the Internet side of my Internet Service Provider's (ISP's) router in hop 4.
By having multiple layers of NAT, the Internet has been able to push off adoption of IPv6 for over thirty years.
Most ISPs isolate their clients behind multiple layers of NAT routers. Most ISPs will allow businesses to pay to have a public IP, but it's difficult if not impossible to buy one unless you have a business.

The only reason someone would need a public IP address is if they want to host an Internet service in their company or home.
Suppose I wanted to create a website. I don't want to pay any company to host my website, I would rather buy the equipment and run it from my home. For people on the Internet to access my website, their computers will contact DNS. DNS needs to be able to give an IP address on the Internet that the can use to contact my web server.
If I'm behind even one layer of NAT, it will be impossible. Remember, the NAT router tags the outgoing data with a port that can be used to route the reply to the original sender. If people start contacting my NAT server asking for the web page, the NAT server doesn't know where to send that traffic because it's not a reply. Instead, it will just drop the request.
To host a service behind a NAT router, you can configure port forwarding. With port forwarding, you tell the NAT router who to send traffic to if it comes in with a particular port number. In my example, let's say my web page uses HTTP. HTTP uses port 80. I could tell the NAT router to send all the traffic coming into port 80 to the server that I'm using to host my website.

IPv6

Background

IP version 6, or IPv6, the successor to IPv4, is an addressing scheme that increases the available pool of IP addresses. IPv6 addresses are 128 bits in binary. IPv6 also includes new features. But to fully implement IPv6 will require a general conversion of IP routers. As of 2023, approximately 50% of the Internet supports IPv6.
IPv6 addresses are written in eight blocks of four hexadecimal numbers. Here's a typical IPv6 address:
2003:a12f:0000:0000:0000:0000:0000:0a12
If there are leading zeros, they can be left out when writing the address. We could rewrite that address as:
2003:a12f:0:0:0:0:0:a12
If there are a number of blocks that are all zeroes, you can replaced them with a double colon. The devices understand that the missing blocks are all zeroes in between the two colons. This can only be used once in the IP address.; We could also rewrite that address as:
2003:a12f::a12
For example, the loopback address in IPv6 is 0:0:0:0:0:0:0:1 but it is always written as ::1.

New Features

In IPv6, address blocks are automatically assigned hierarchically by routers. Top-level routers have top-level address blocks. These are automatically divided and assigned as routers. Segments are added to the address blocks. This divides the address space logically instead of randomly, making it easier to manage.
A new field in the IP header of IPv6 packets enables IP to guarantee the allocation of network resources when requested by time-dependent services such as voice and video transmission.
IPv6 has built-in support for IPSec. That means it offers built-in encryption.

Unicast Address Structure

IPv6 replaces classful addresses with a more flexible and logical addressing structure. There are different categories of unicast addresses that serve different functions. Each network interface on a typical IPv6 host will be logically multihomed. Multihomed either means more than one NIC or more than one IP address. As it relates to IPv6, it means that IPv6 devices will have more than one type of unicast address assigned. There are four types of IPv6 addresses:

IPv4 Address Type	IPv6 Equivalent	IPv6 Address Starts with:
Public	Global Unicast	2 or 3
Private	Site-Local	FC or FD
APIPA	Link-Local	Fe8
Multicast	Multicast	FF

IPv6 works on all the same rules as IPv4. The network address is called the prefix. By default, it is the first half (64 bits) of the IPv6 address.
IPv6 has two addressing modes for dynamic clients: stateless and stateful.
In stateless addressing, the client gets the prefix for the network from the router. It uses the IPv6 MAC Address (EUI-64) which is 64 bits as the host address. Since the EUI-64 is guaranteed to be unique, that gives the client a unique IP address with the right prefix.
In stateful addressing, the IPv6 client gets the IP address from a DHCP server.

Internet Protocols Lab

Design and Implement an IPv4 Network

Explore NICs

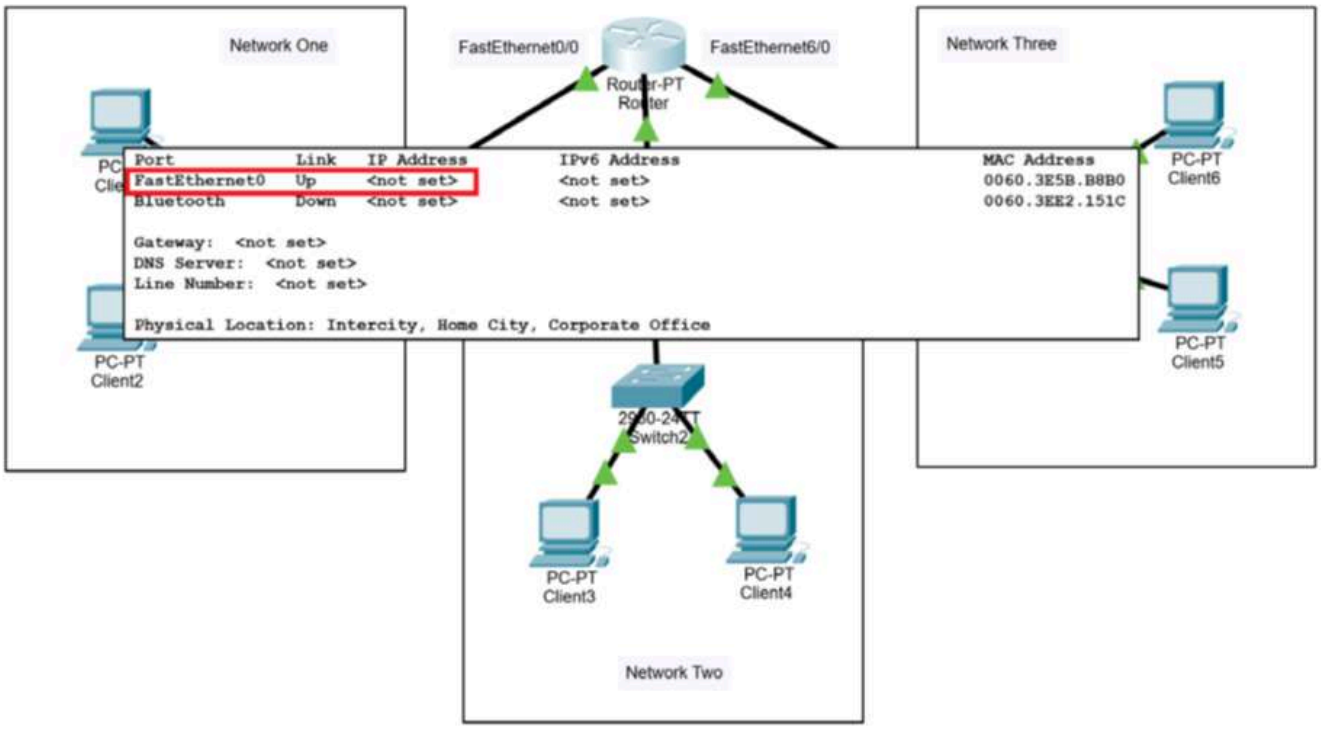
In this lab, you will design and implement an IPv4 network. Note: unlike previous labs, this lab will not provide a lot of screenshots. If you cannot remember how to do something, refer to the pictures in the previous labs.
There are several correct answers to this lab. The steps in the lab are provided to guide you through designing and configuring a network. You have successfully completed the lab when every client can connect to every other client.

TASK A

In this activity you will design an IP addressing scheme for the network. Your addressing scheme must meet the following requirements:

1. The addresses you use must all be Class C addresses.
2. The addresses you use must be private IP addresses.
3. All three networks should use the same subnet mask.
4. The subnet mask you choose should allow for at least three networks or at least 10 clients on each network.

1. Download the **4.4.1 Lab File** and open it in **Packet Tracer**.
[4.4.1 Lab File]
[PKT File]
2. Hover your mouse over all six of the clients and the router. Notice that none of the devices has been assigned an IP address or subnet mask.



3. Record the subnet mask you intend to use in your networks: __255.255.255.0_____
4. Record the network IDs you plan to use:
   a. Network One: ___192.168.1.0_____
   b. Network Two: ____192.168.2.0_____
   c. Network Three: _____192.168.3.0_____
5. Record the IP addresses you plan to use on **Network One**:
   a. Client1: _____192.168.1.10_____
   b. Client2: _____192.168.1.11_____
   c. Router Fa0/0: _____192.168.1.1_____
6. Record the IP addresses you plan to use on **Network Two**:
   a. Client3: _____192.168.2.10_____
   b. Client4: _____192.168.2.11_____
   c. Router Fa1/0: _____192.168.2.1_____
7. Record the IP addresses you plan to use on **Network Three**:
   a. Client5: _____192.168.3.10_____
   b. Client6: _____192.168.3.11_____
   c. Router Fa6/0: __192.168.3.1_____

TASK B

In this task, you will configure the network.
1. Click on **Client1** to open the **Client1 Properties** dialog box.
2. On the **Config** tab, in the **Interface** menu, select **FastEthernet0**.

3. In the **IPv4 Address** text box, type the address you have assigned to Client1.
4. In the **Subnet Mask** text box, type your subnet mask.
5. On the **Config** tab, in the **Global** menu, select **Settings**. In the **Gateway/DNS IPv4** section, in the **Default Gateway** text box, enter the correct address.
6. Repeat these steps for the remaining clients.
7. Click on **Router** to open the **Router Properties** dialog box.
8. On the **Config** tab, in the **Interface** menu, select **FastEthernet0/0**.
9. Enter the IP address you have assigned to the router on Network One.
10. Select **FastEthernet1/0**.
11. Enter the IP address you have assigned to the router on Network Two.
12. Select **FastEthernet6/0**.
13. Enter the IP address you have assigned to the router on Network Three.

TASK C

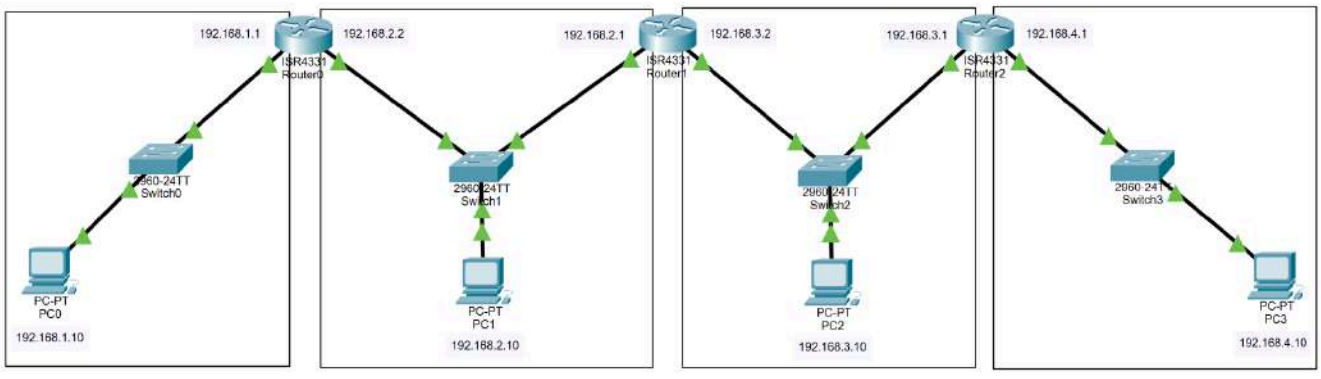
In this task, you will test your configuration.
1. Click on **Client1** to open the **Client1 Properties** dialog box.
2. In the **Desktop** tab, click the **Command Prompt** icon.
3. In the **Command Prompt**, use the **ping** command to verify connectivity to all the other computers in the network.
4. Repeat these steps from the remaining clients to test your configuration.
If all the clients can ping each other, you have completed the lab successfully. If something doesn't work, check all the devices in question. Use basic ANDing to make sure you have a good design. If your design is correct, look carefully to be sure there are no typing mistakes.

Configure Routing

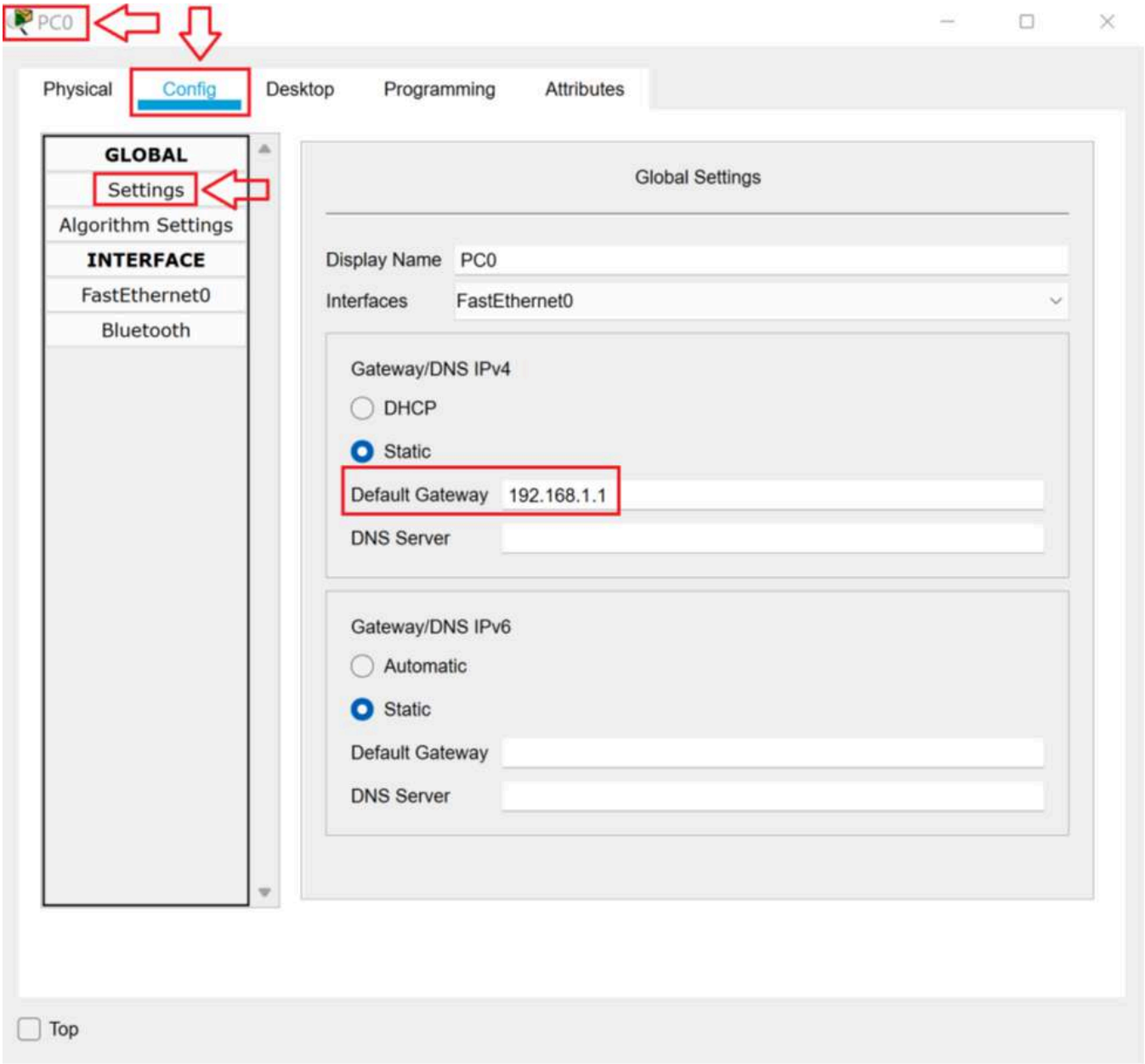
n this lab, you will configure routing. The lab is complete when all four of the clients can ping each other by IP address.

TASK A

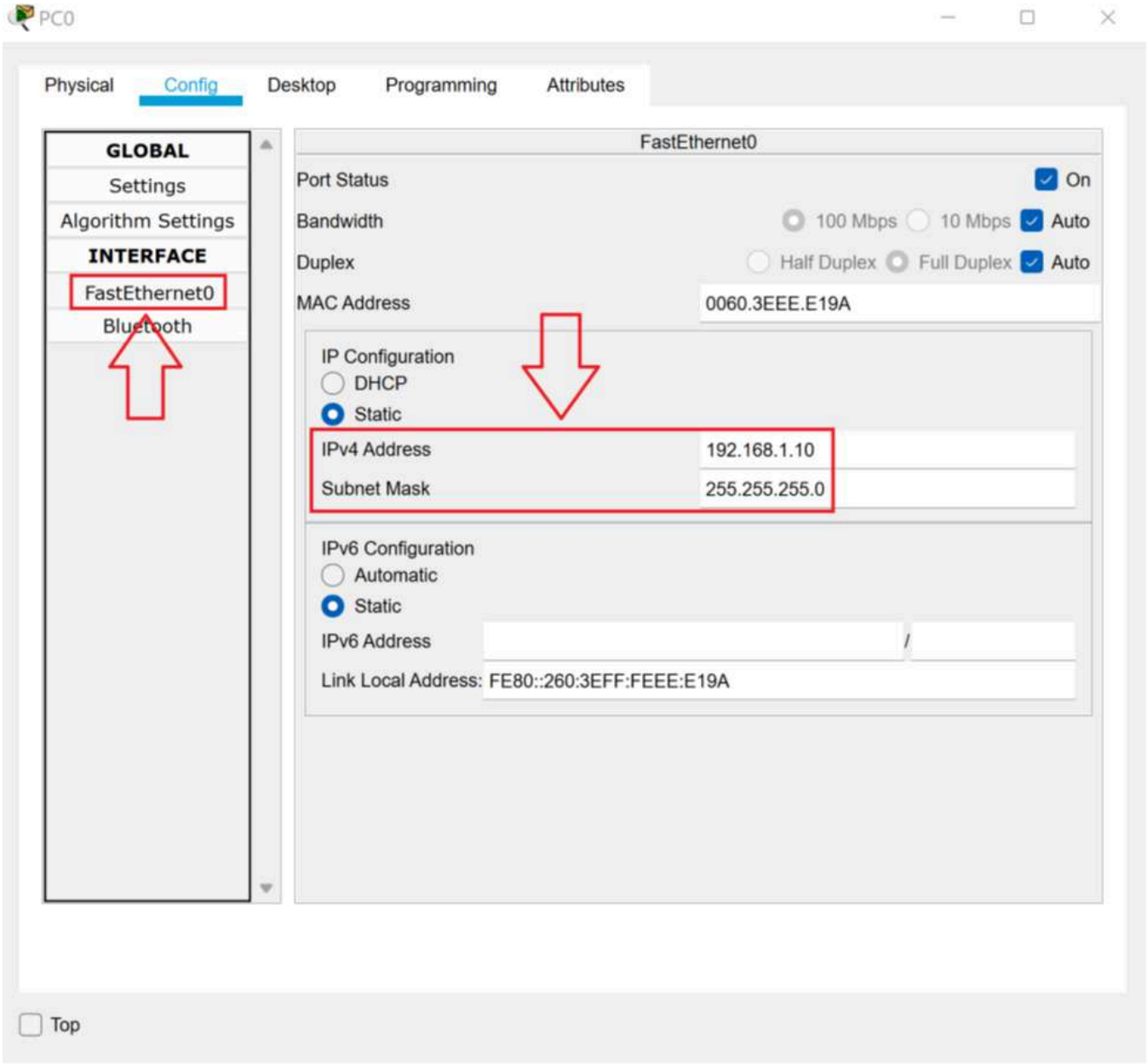
In this task, you examine the network setup.
1. Download the **4.4.2 Lab File** and open it in **Packet Tracer**.
4.4.2 Lab File
PKT File



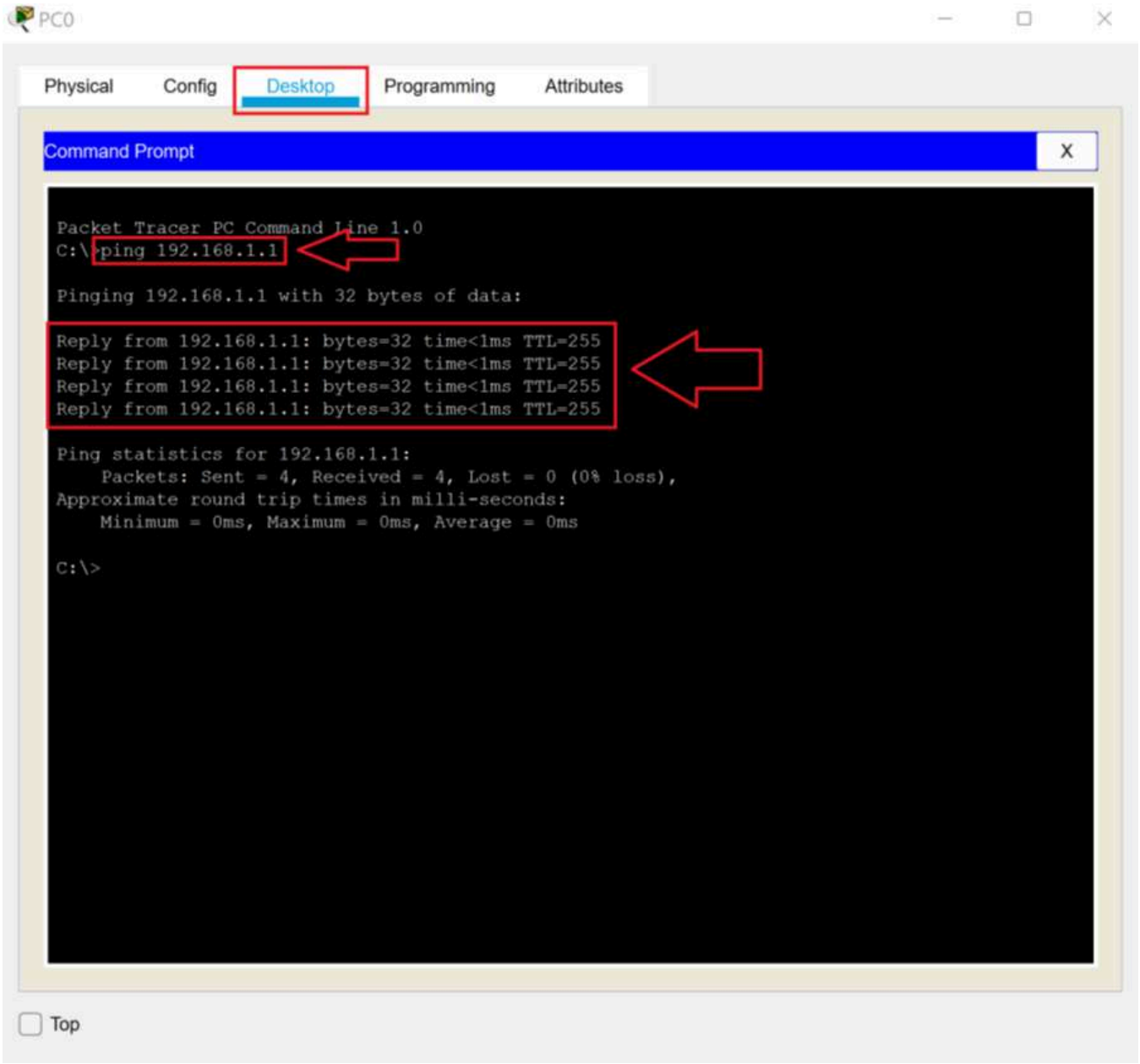
2. Click **PC0** to open the **PC0 Properties** dialog box. On the **Config** tab, in the **Global Settings** menu, observe the **Default Gateway**.



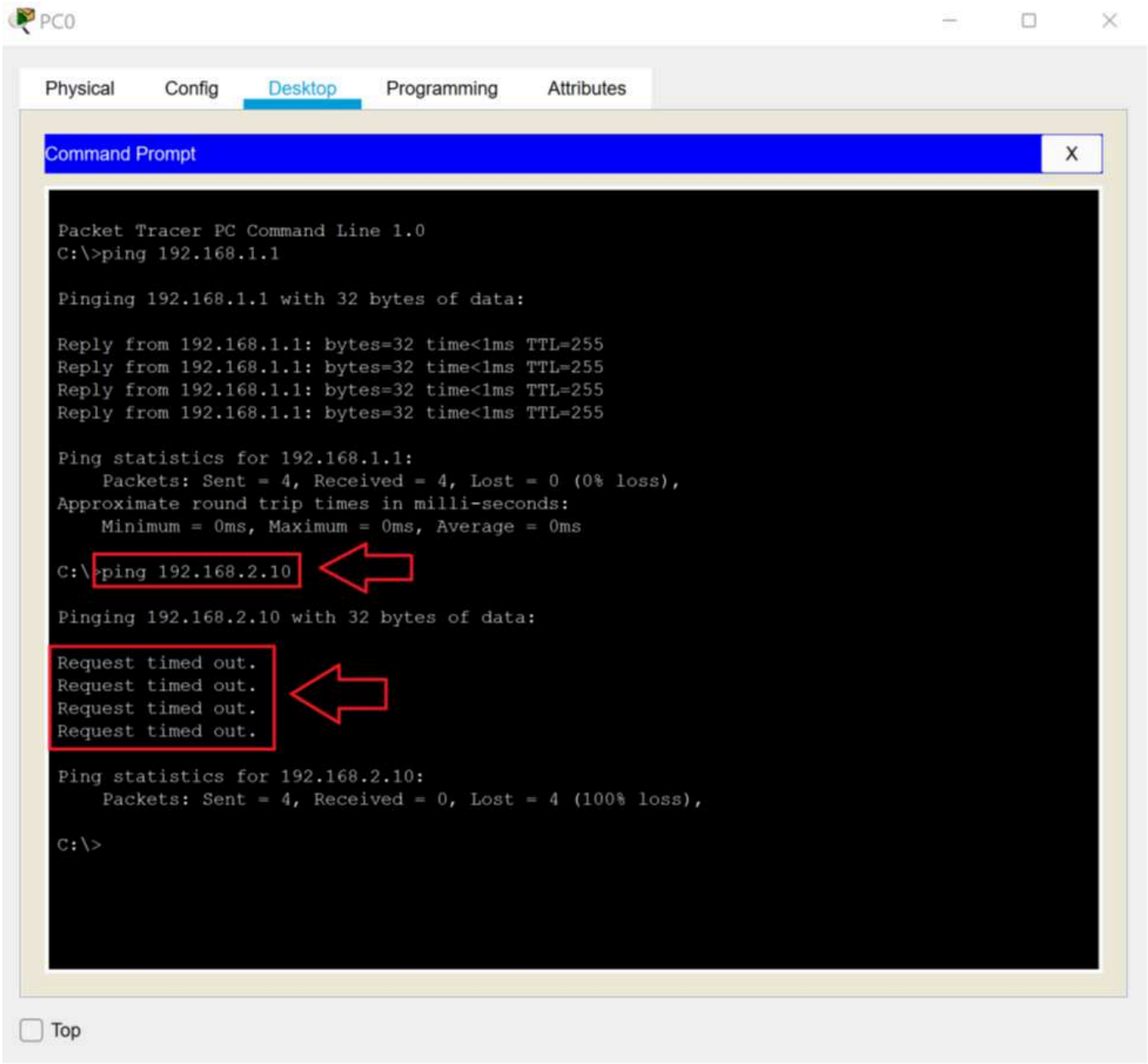
3. Click the **FastEthernet0** menu. Observe the client **IPv4 Address** and **Subnet Mask**.



4. Click the **Desktop** tab. Click the **Command Prompt** icon. In the command prompt, type **ping 192.168.1.1** and press **Enter**. Notice the client receives four replies.



5. In the **Command Prompt**, type **ping 192.168.2.10** and press **Enter**. Notice the request timed out.



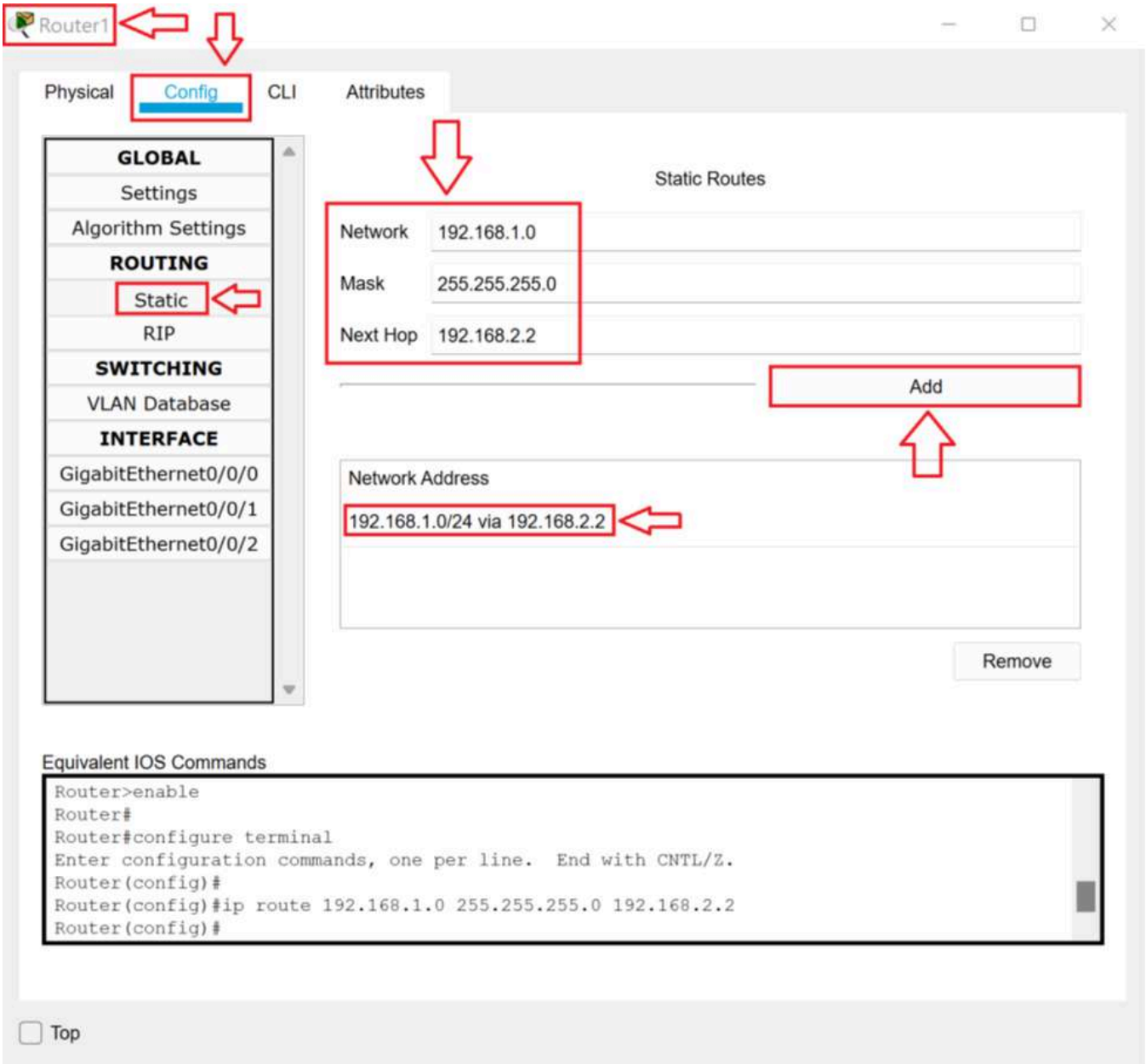
6. In the **Command Prompt** use the **ping** command to ping **192.168.3.10** and **192.168.4.10**. Notice the Destination Host Unreachable message.
7. Close the **PC0 Properties** dialog box.
8. Follow the same procedure to examine the Default Gateway, IPv4 Address and Subnet Masks of the other three clients. Verify that each of the clients can ping its default gateway but that the request times out or they get Destination Host Unreachable to all other clients except that 192.168.3.10 can ping 192.168.4.10 and vice versa.

TASK B

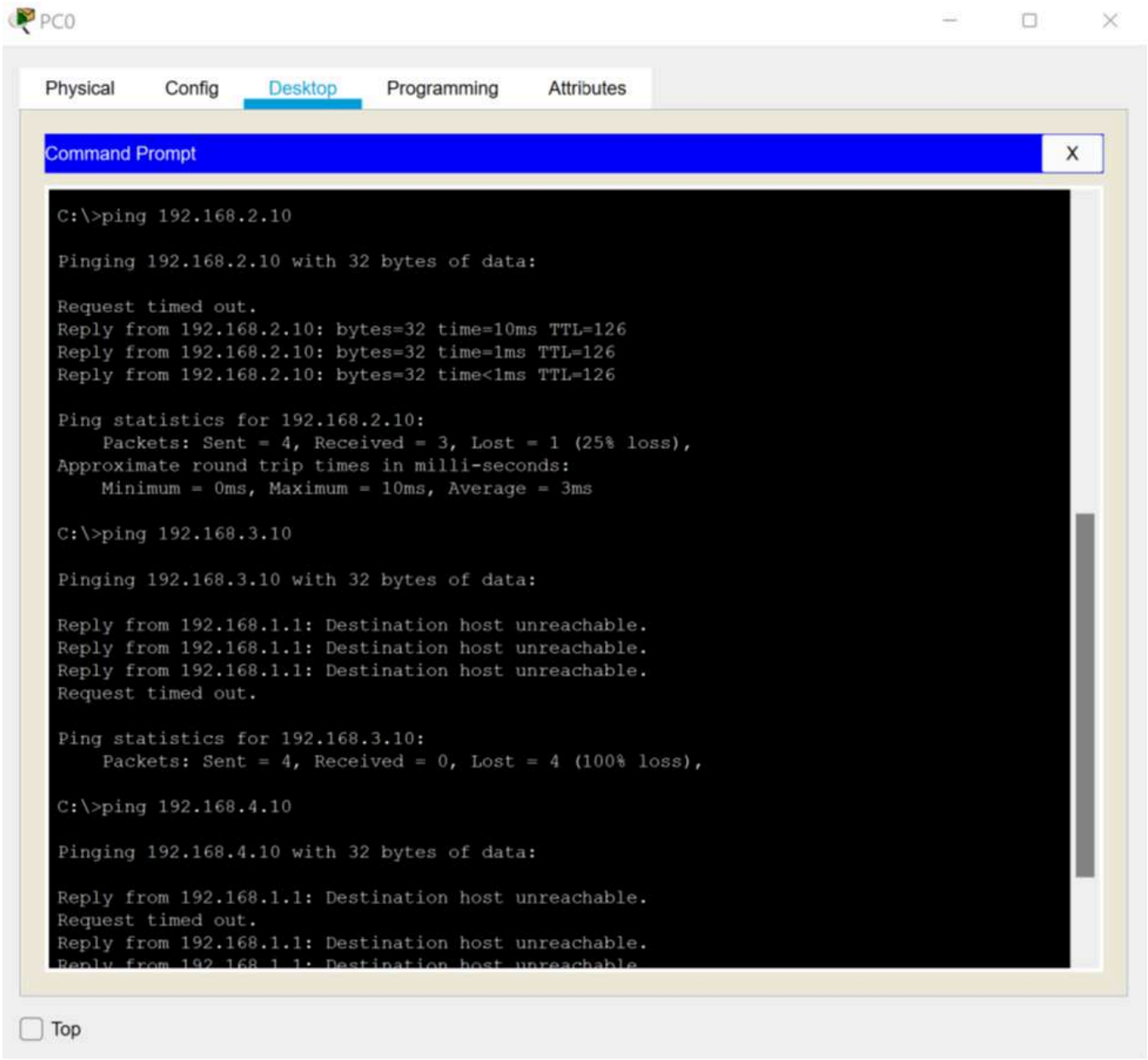
In this task, will resolve the routing problem between the 192.168.1.0 /24 network and the 192.168.2.0 /24 network. When PC0 pings PC1, PC0 sends the ping to Router0 at 192.168.1.1. Router0 sends the ping out its 192.168.2.2 interface. The ping arrives at PC1. PC1 sends the reply to Router1 at 192.168.2.1. However, Router1 does not have an interface on the 192.168.1.0 /24 network or an entry

in its routing table. You must configure Router1 to send data for the 192.168.1.0 /24 network to Router0.

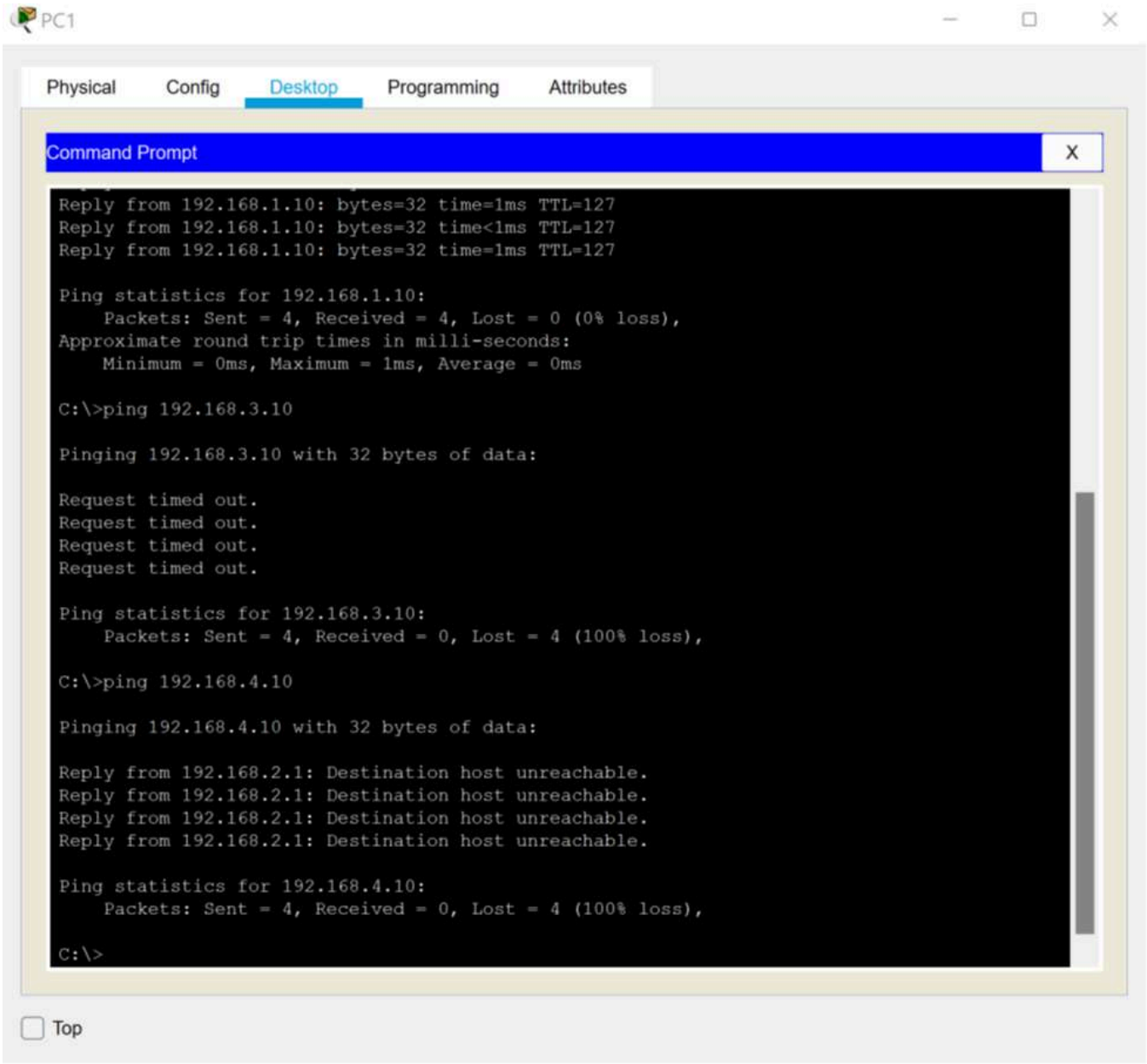
1. Click **Router1** to open the **Router1 Properties** dialog box. On the **Config** tab, click **Static**. In the **Network** text box, type **192.168.1.0**. In the **Mask** text box, type **255.255.255.0**. In the **Next Hop** text box, type **192.168.2.2**. Click the **Add** button. Notice the route has been added. Then close the **Router1 Properties** dialog box.



2. On **PC0**, verify that **PC0** can now ping **PC1**. All other clients still get Destination Host Unreachable.



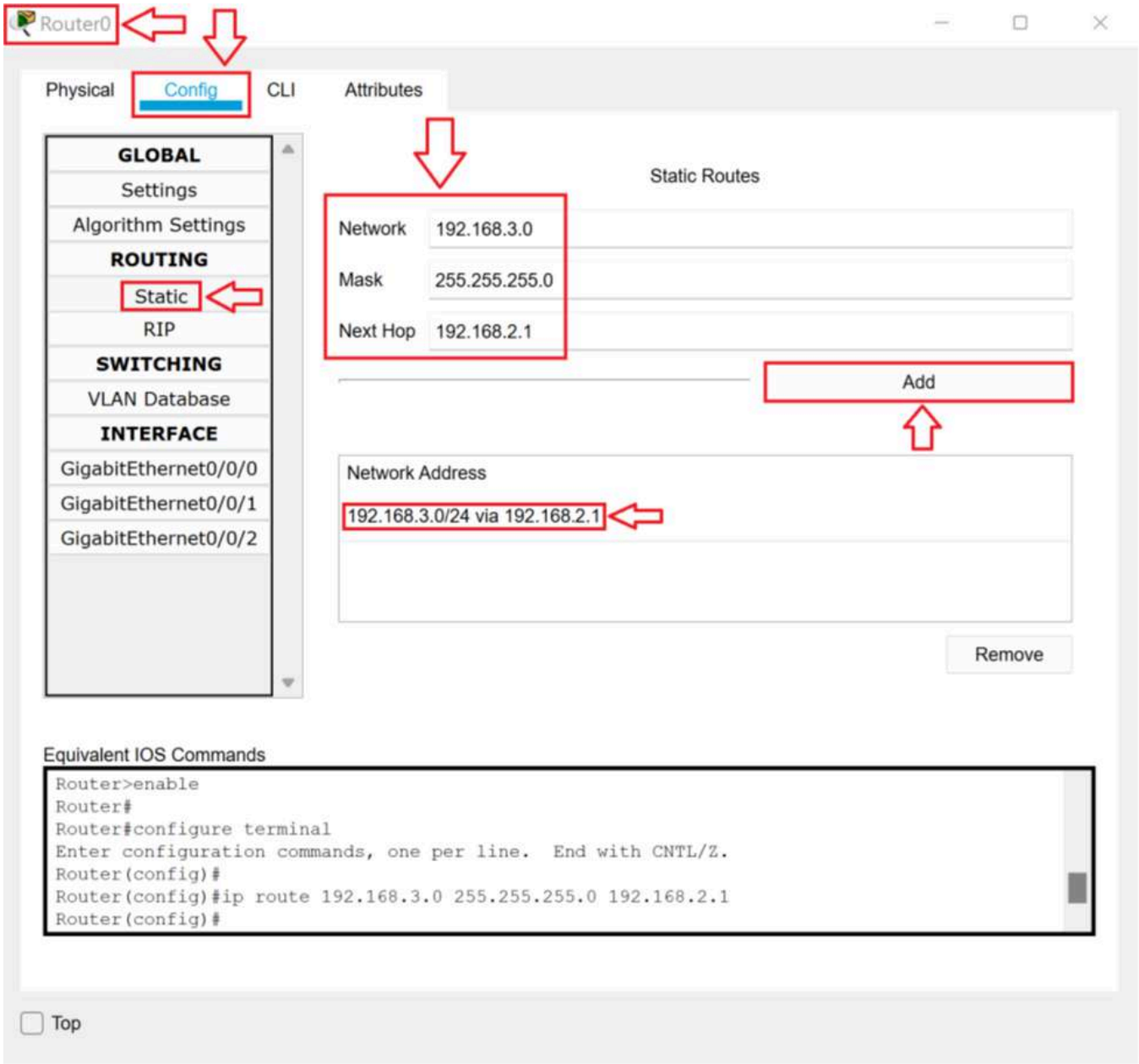
3. On PC1, verify PC1 can now ping PC0 but it still cannot ping PC2 or PC3.



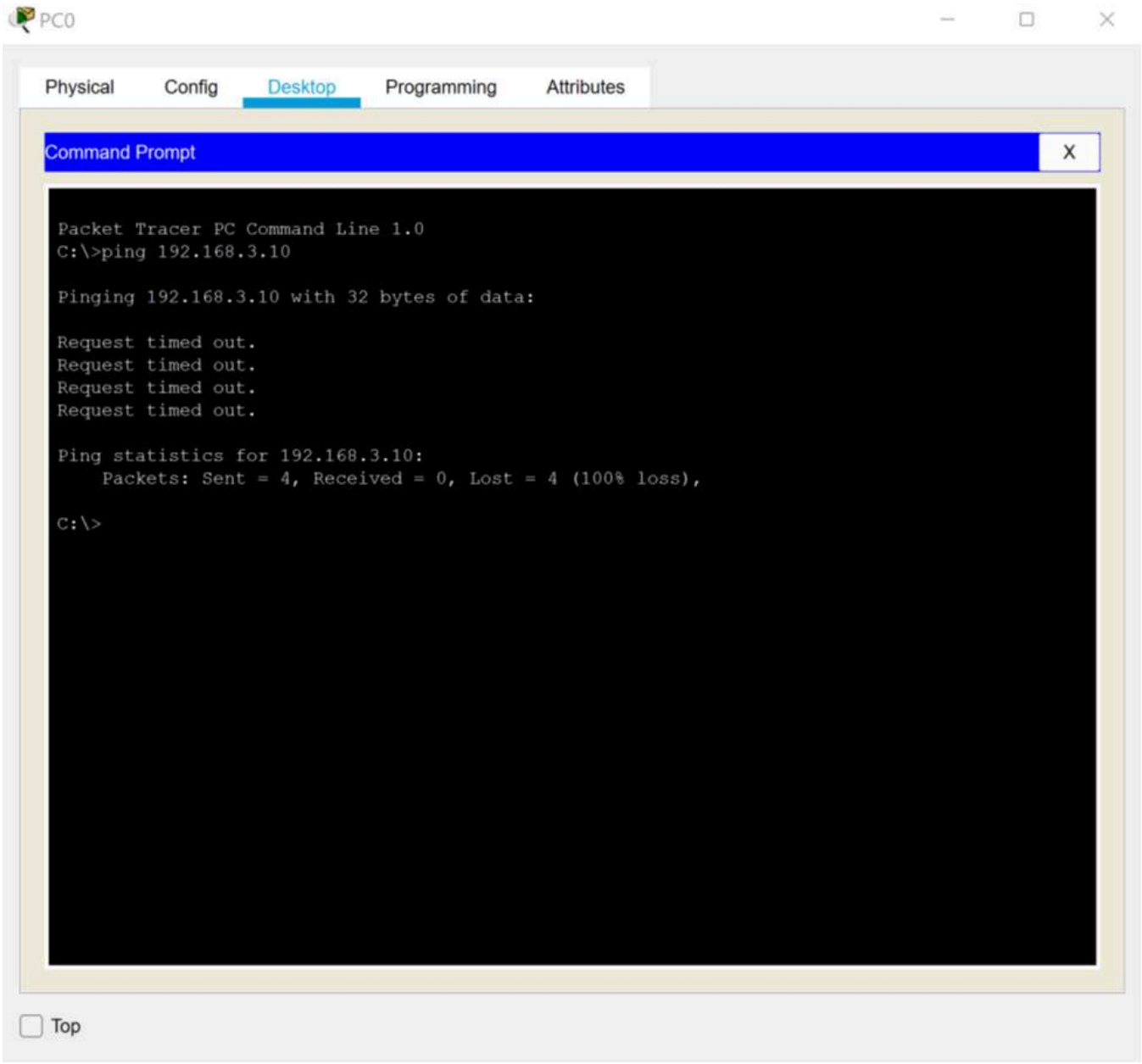
TASK C

In this task, will resolve the routing problem between the 192.168.1.0 /24 network and the 192.168.3.0 /24 network. When PC0 pings PC2, PC0 sends the ping to Router0 at 192.168.1.1. Router0 does not have an interface in the 192.168.3.0 /24 or an entry in its routing table. You must start by making an entry in the Router0 routing table, so Router0 knows where to send the data.

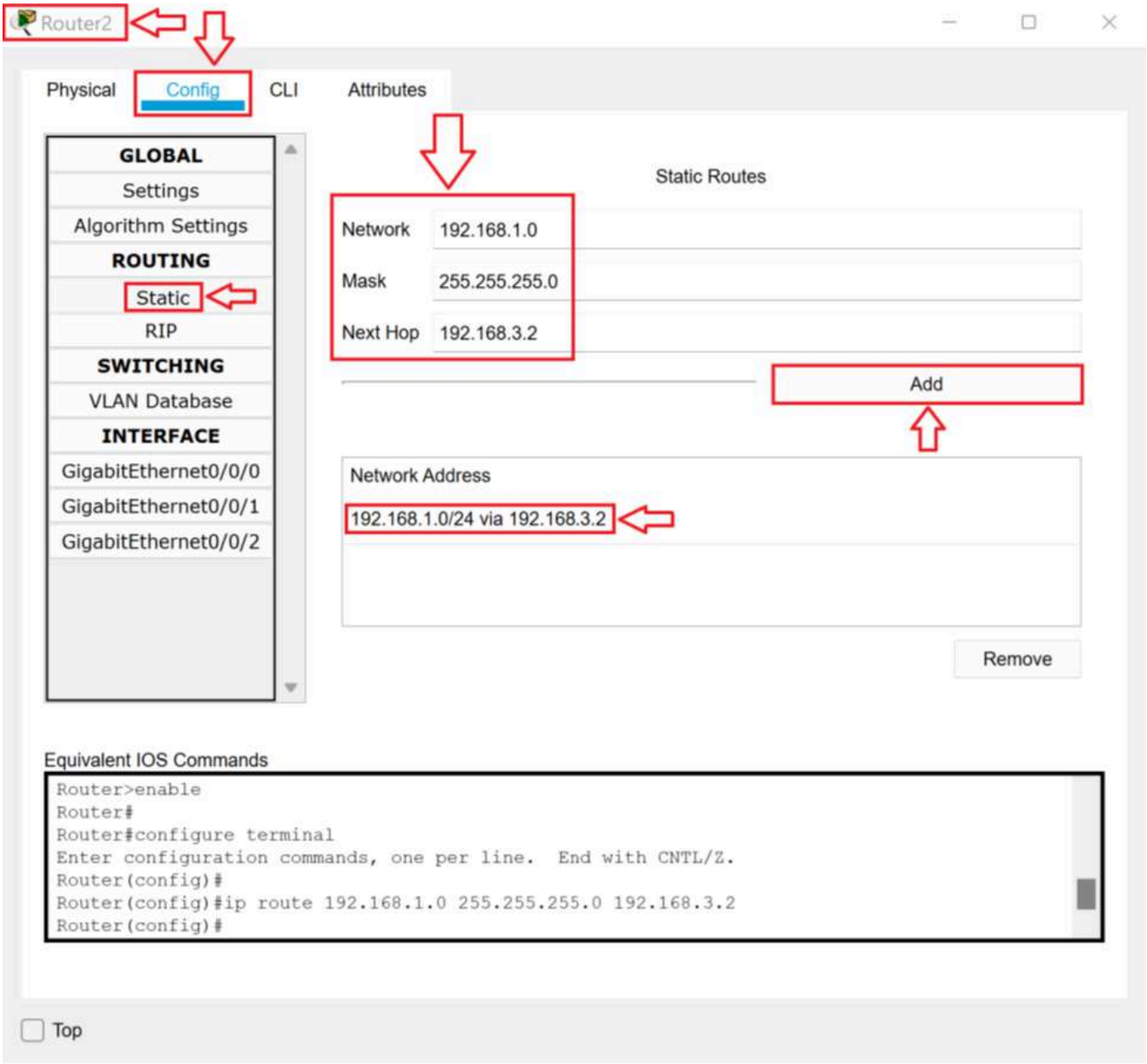
1. Click **Router0** to open the **Router0 Properties** dialog box. On the **Config** tab, click **Static**. In the **Network** text box, type **192.168.3.0**. In the **Mask** text box, type **255.255.255.0**. In the **Next Hop** text box, type **192.168.2.1**. Click the **Add** button. Notice the route has been added. Then close the **Router0 Properties** dialog box.



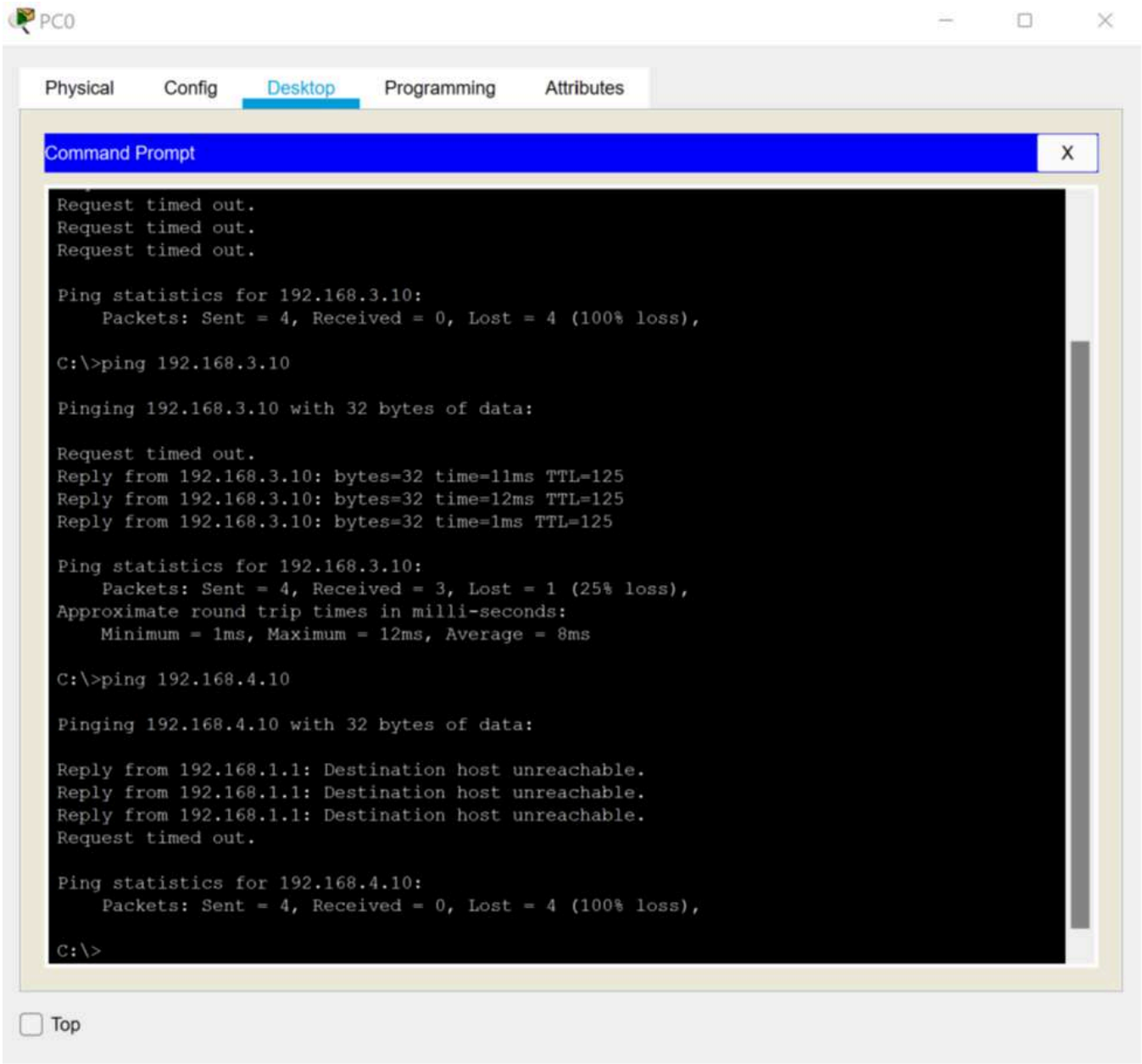
2. On **PC0**, verify that the client no longer gets a Destination Host Unreachable error. Now it gets a Request Timed Out. When **PC0** pings **PC2**, **PC0** sends the ping to **Router0** at 192.168.1.1. Now **Router0** sends the data to **Router1** at 192.168.2.1. **Router1** has an interface on the 192.168.3.0 /24 network and sends the data to **PC2**. **PC2** sends the reply to **Router2** at 192.168.3.1. **Router2** does not have an interface in the 192.168.1.0 /24 network, nor does it have an entry in its routing table. **Router2** needs a routing entry for the 192.168.1.0 /24 network.



3. Click **Router2** to open the **Router2 Properties** dialog box. On the **Config** tab, click **Static**. In the **Network** text box, type **192.168.1.0**. In the **Mask** text box, type **255.255.255.0**. In the **Next Hop** text box, type **192.168.3.2**. Click the **Add** button. Notice the route has been added. Then close the **Router2 Properties** dialog box.



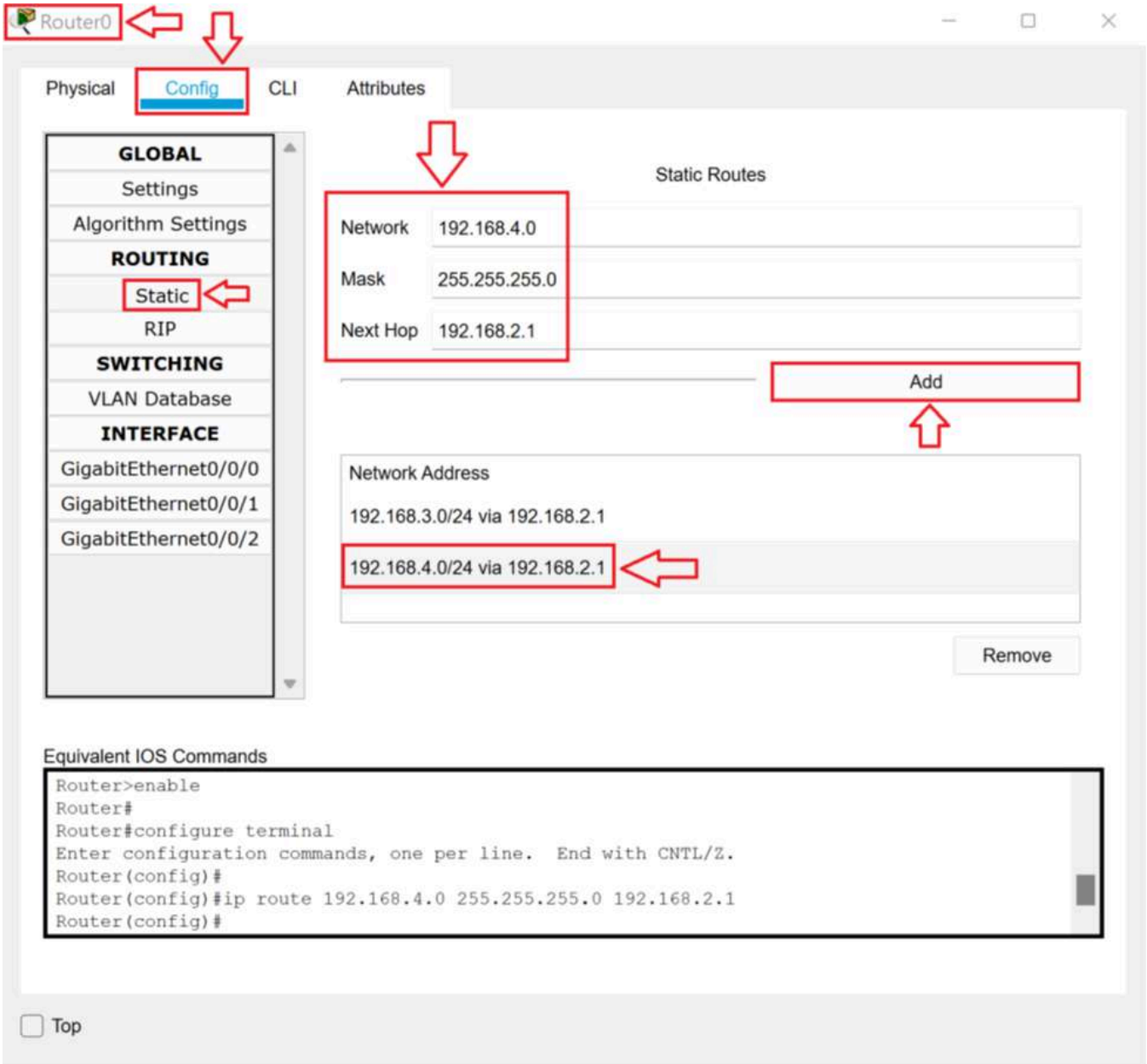
4. Verify that **PC0** can now ping **PC2**. It still gets a Destination Host Unreachable reply when pinging **PC3**.



TASK D

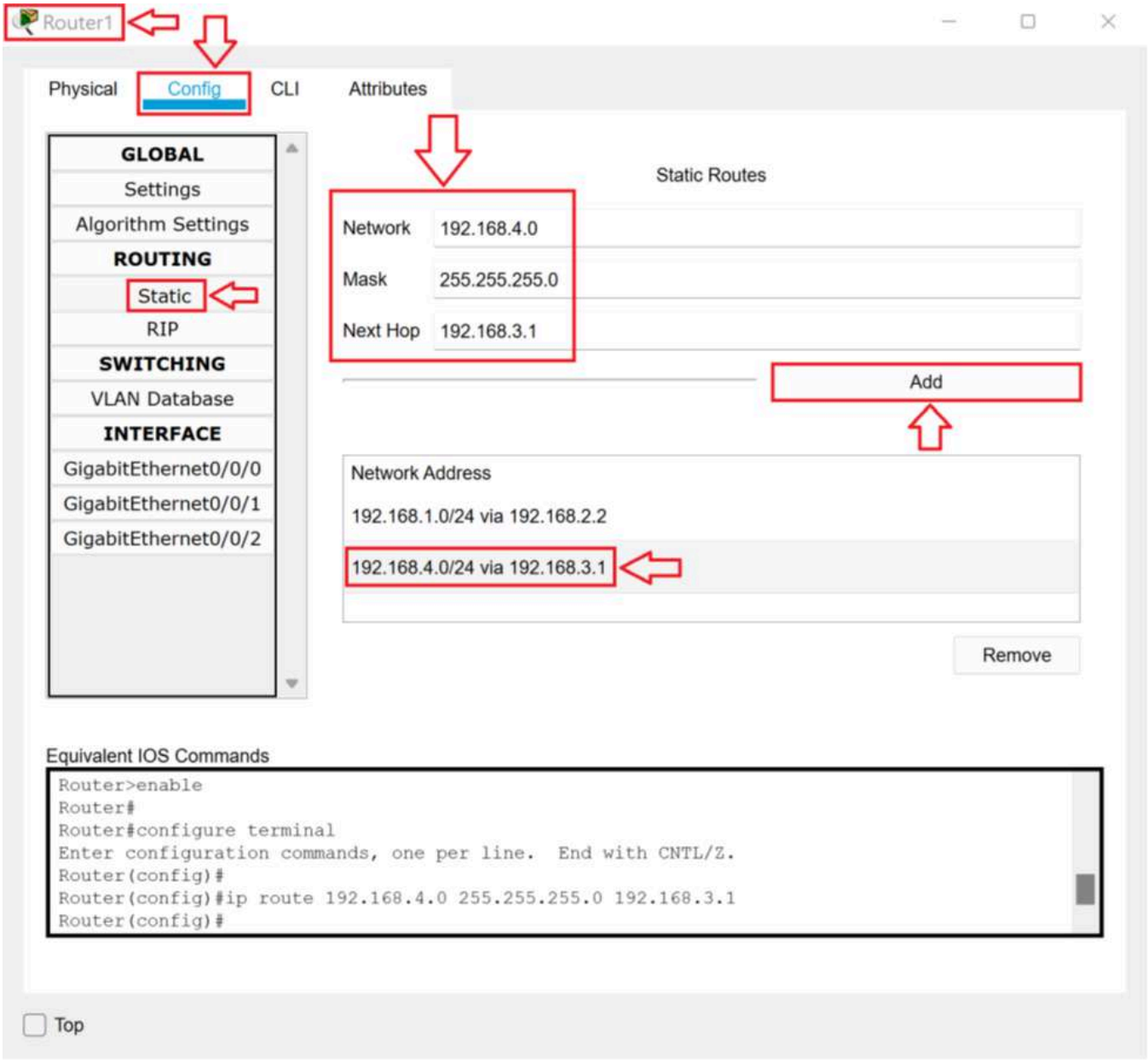
In this task, will resolve the routing problem between the 192.168.1.0 /24 network and the 192.168.4.0 /24 network. When PC0 pings PC3, PC0 sends the ping to Router0 at 192.168.1.1. Router0 does not have an interface in the 192.168.4.0 /24 or an entry in its routing table. You must start by making an entry in the Router0 routing table, so Router0 knows where to send the data.

1. Click **Router0** to open the **Router0 Properties** dialog box. On the **Config** tab, click **Static**. In the **Network** text box, type **192.168.4.0**. In the **Mask** text box, type **255.255.255.0**. In the **Next Hop** text box, type **192.168.2.1**. Click the **Add** button. Notice the route has been added. Then close the **Router0 Properties** dialog box.



2. On PC0, confirm that now PC0 still gets a Destination Host Unreachable message when pinging PC3. When PC0 pings PC3, PC0 sends the ping to Router0 at 192.168.1.1. Router0 now sends the data to Router1. Router1 does not have an interface in the 192.168.4.0 /24 or an entry in its routing table.

3. Click **Router1** to open the **Router1 Properties** dialog box. On the **Config** tab, click **Static**. In the **Network** text box, type **192.168.4.0**. In the **Mask** text box, type **255.255.255.0**. In the **Next Hop** text box, type **192.168.3.1**. Click the **Add** button. Notice the route has been added. Then close the **Router1 Properties** dialog box.



4. Verify that PC0 can now successfully ping PC1, PC2, and PC3.
5. Close the **4.4.2 Lab File** file. You do not need to save the changes.

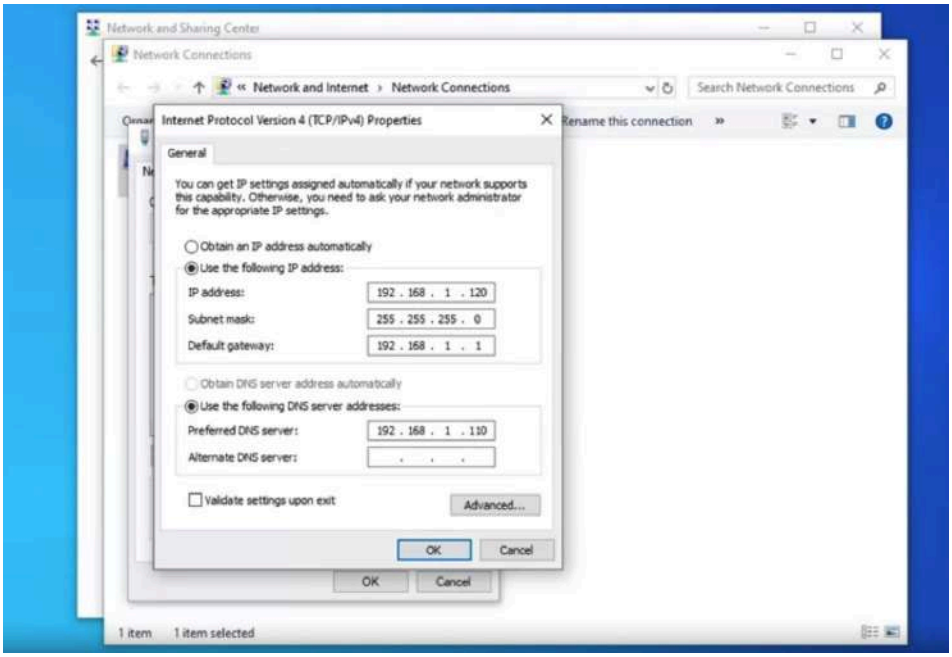
Week 3

TCP/IP Services

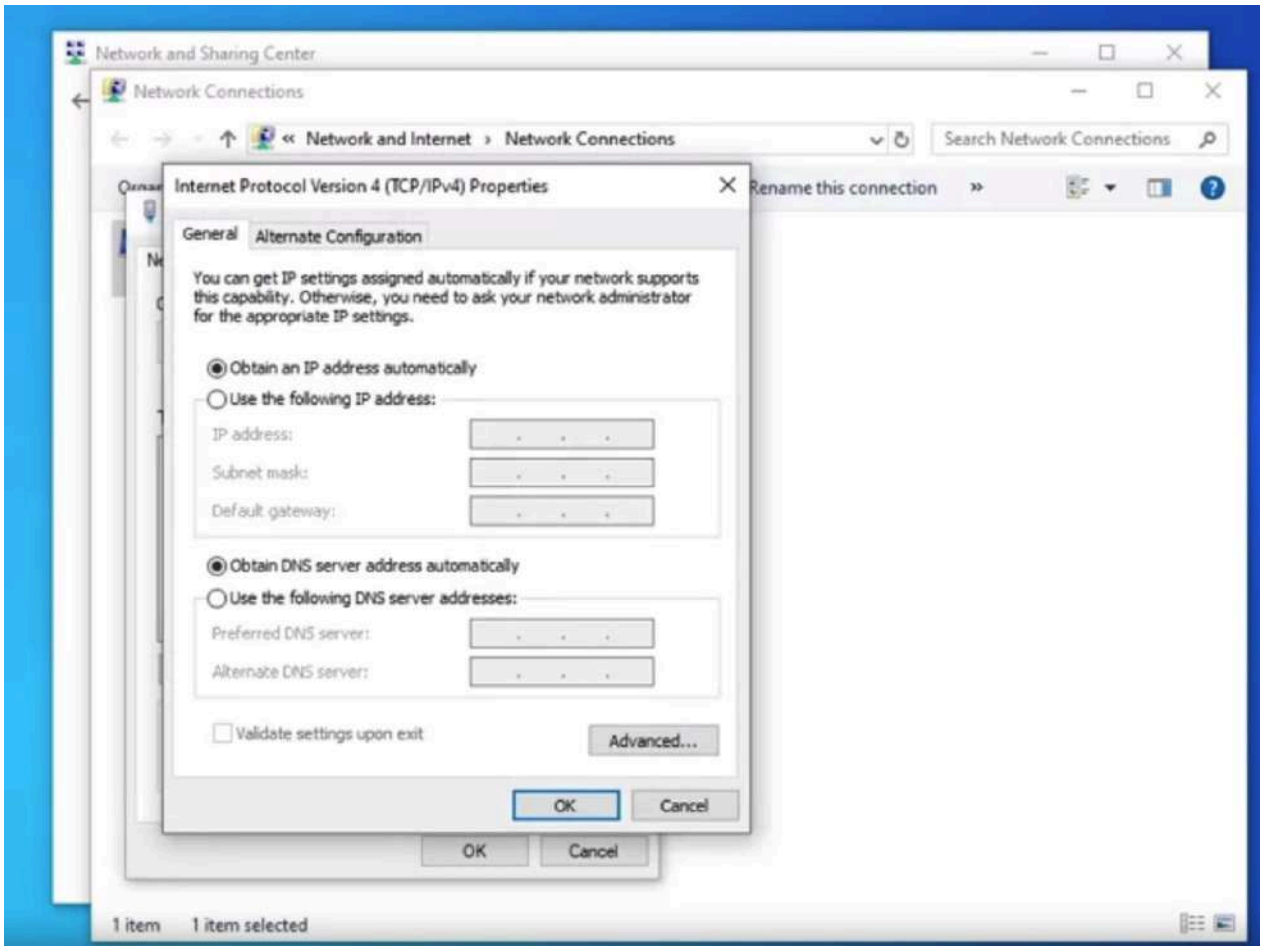
Assign IP Address

Static and Dynamic Addressing

When an administrator types in an IP address and a subnet mask directly into the client. As you can see I've done here that address is called a static IP address. It's called static because the device won't change the address until the administrator types and new one. Now static IP addresses should be used for any device that needs to have an address, even if DHCP is not working and that address shouldn't change. Typically these are critical infrastructure servers like the DHCP servers, DNS servers, directory servers. The DHCP server itself always must have a static IP address.

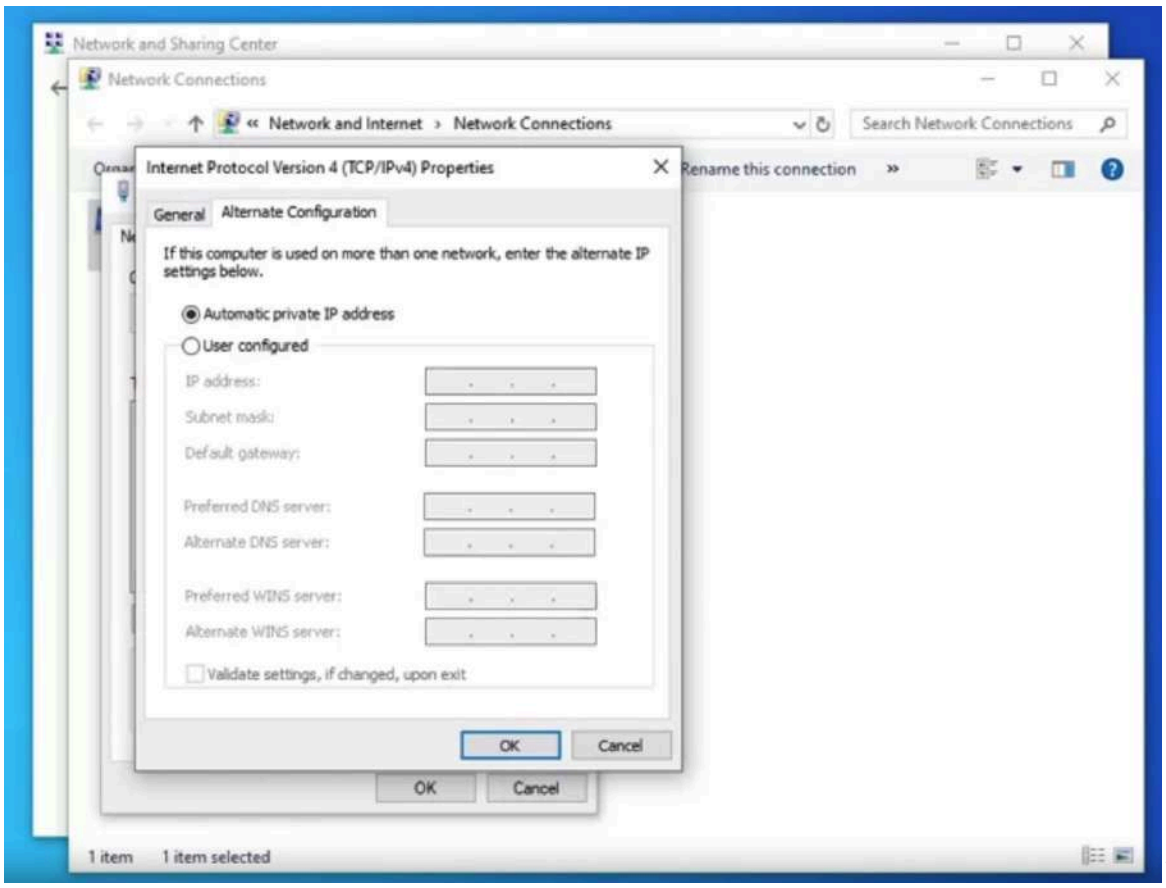


The only disadvantage of a static IP address is that if the IP design of the network needs to change or if the server needs to be moved to a different network, the administrator will need to manually change the IP address. But I'm on a client machine if this was a laptop, and I took this to another location, like let's say I'm home and now I go out to a coffee shop, then I would have to come in here and change this. When a client obtains an IP address via DHCP, that address is called a dynamic address and Microsoft calls it, obtain an IP address automatically. But once I click in that radio button, it's telling the client you should go to DHCP to get your address. It's called dynamic because the client can only use the IP address for a short period of time. The amount of time that the client can use the IP address is called the lease. Then when the lease expires, the client can renew that lease. If the client is unable to renew the lease or the device is moved to a different network, the client will obtain a new IP address from the DHCP server. Dynamic addresses can change at any time depending on what's going on with the DHCP server. The advantage of a dynamic address is the fact that it can change. If I have my laptop at home, I pick up an IP address for my home network. That's great. I go to school, I get a different IP address. That's fantastic. The disadvantage of a dynamic address is that it's dependent on DHCP. If the DHCP server doesn't respond, the client will not have a good IP address on the network and in that case, the client will use an APIPA address.



If you watch, I'm going to click in the static bubble. If you look up here, there's just one tab, general. When I click over to obtain an IP address automatically, you can see that this Alternate Configuration tab pops up. Then this is what's causing my computer to obtain an APIPA address if DHCP doesn't respond. Now if there's a situation where you can anticipate that DHCP is not going to respond and you don't want it an APIPA address. On a Windows machine, you could actually put in an alternate static address to use. But for

everything that you're going to do in your work, you should just assume that people left this because hardly anybody even realizes this is here. I'm going to leave this set if DHCP doesn't talk to you, go for an APIPA address. Let's take a look at what happens to the client. I do have a DHCP server, but I have it turned off so it won't reply to the client so that we can see APIPA. Then we'll go ahead and turn DHCP on and then we'll get the difference. I'm just going to go into a command prompt.

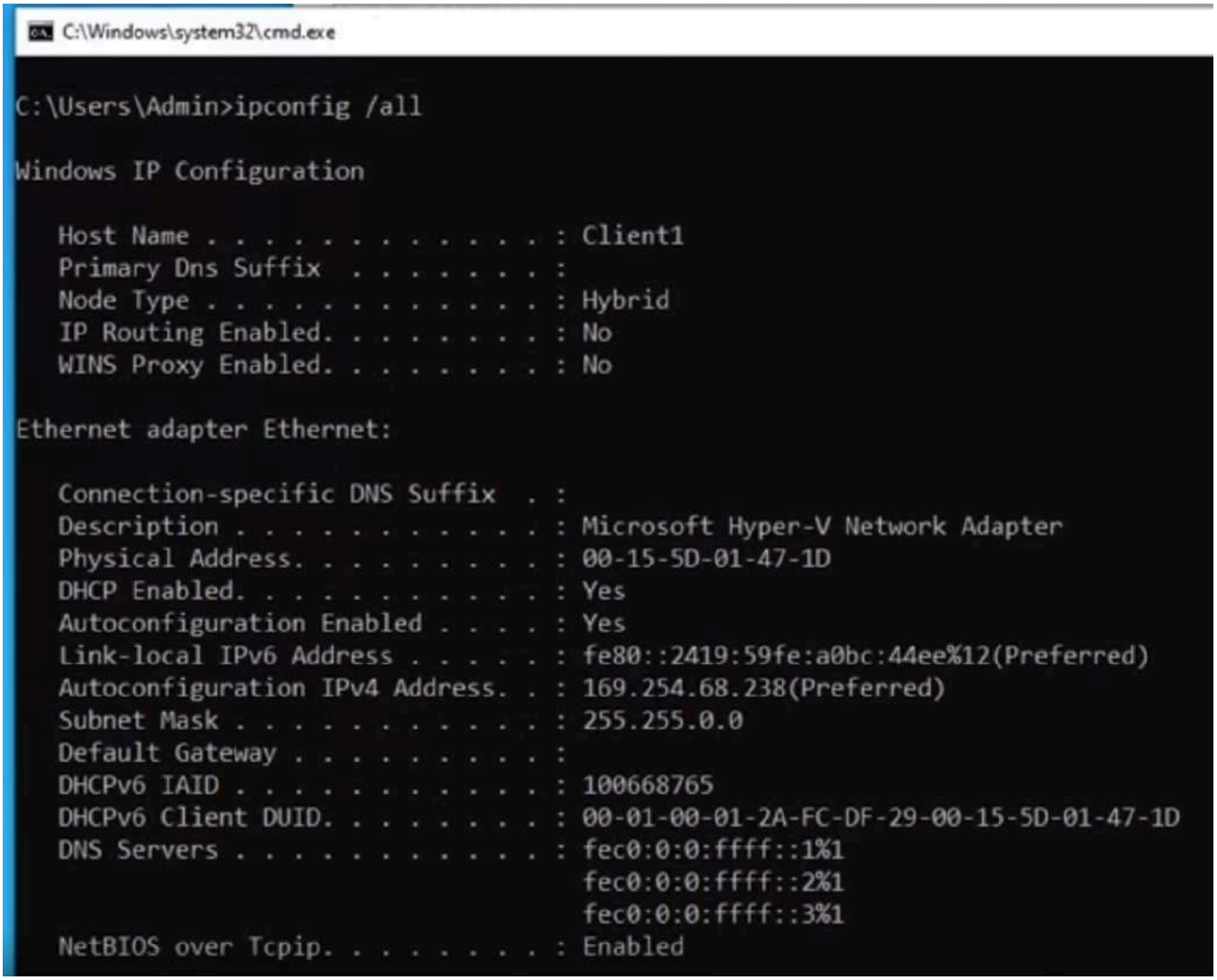


\

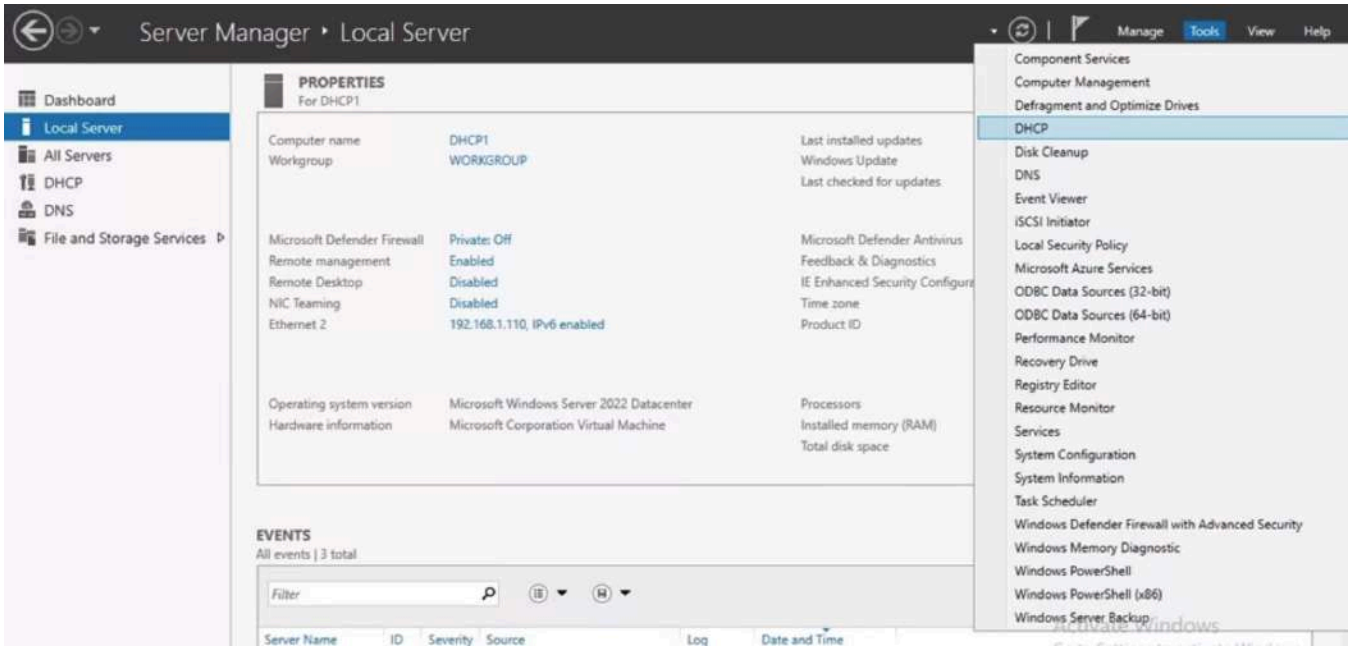
Let's do our IP config. You can see I picked up an APIPA address. We know that because it starts with 169.254 and it's got that 16-bit subnet mask 255.255.0.0. Notice the APIPA addresses don't get a default gateway. If I do IP config all,

we don't really have a DNS server.

We're just getting very limited information. Now, the client is going to try about five times to contact the DHCP server before they set the APIPA address. After that, they periodically retry to contact the DHCP server, but there are longer times between each try until eventually, the client gives up. That means if it takes you a long time to fix the DHCP server, you may need to reboot the clients for them to get an IP address from the DHCP server after you fix it.



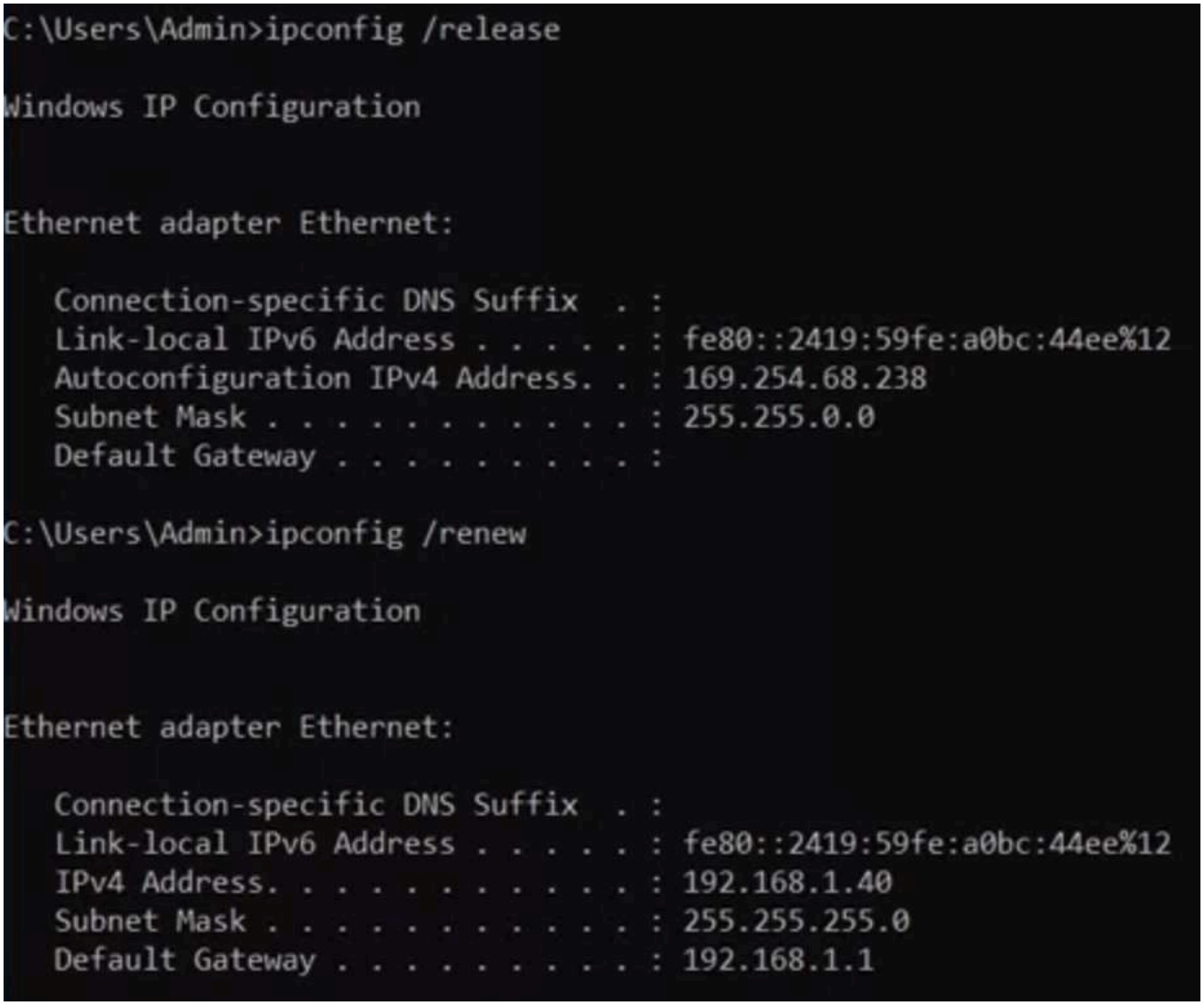
So let's go ahead and fix our DHCP server. I'm going to go over to the DHCP server and let's take a look at what's going on over there. We're here on our DHCP server. It's actually named DHCP1. I'm going to go into the DHCP management console.



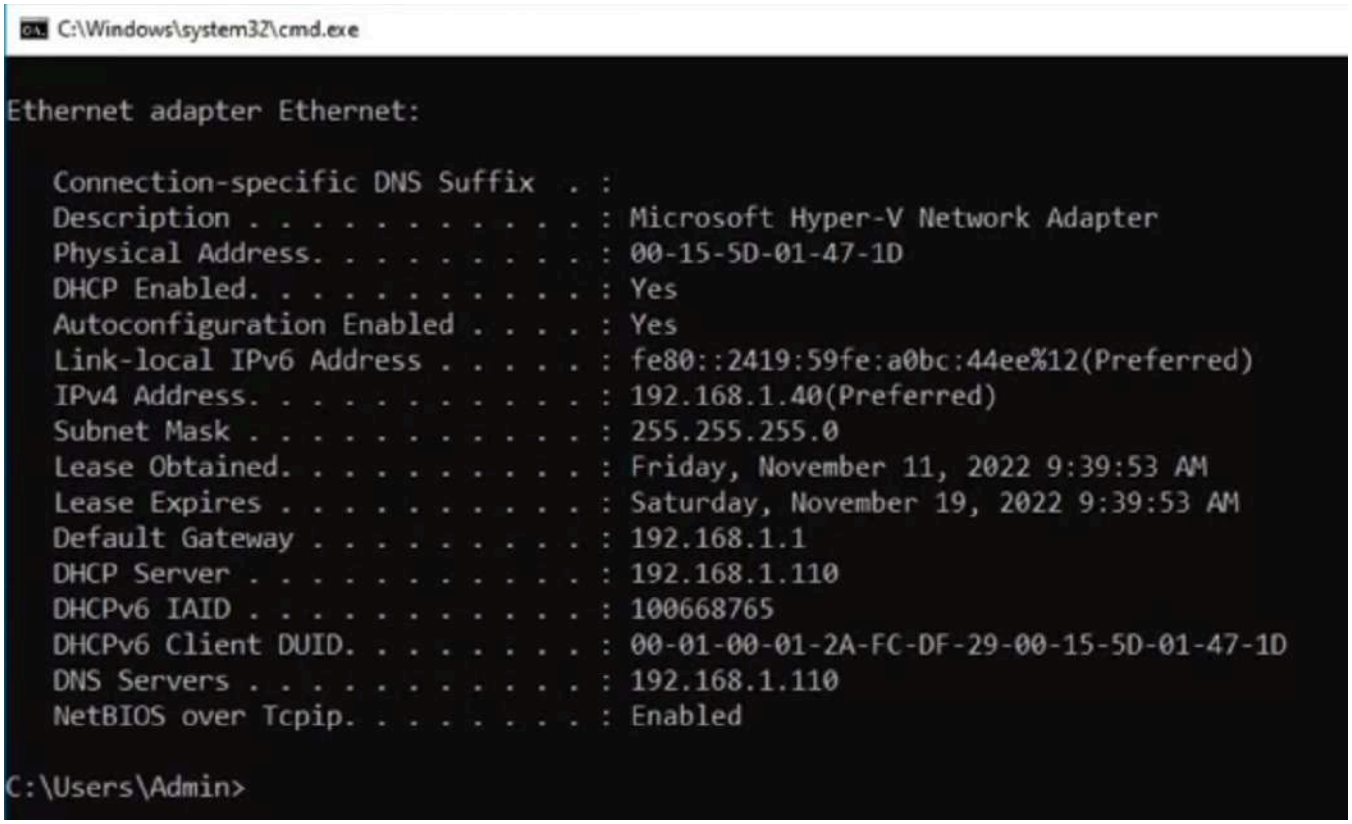
If we take a look with IPv4, the reason my DHCP server hasn't responded is this scope is the pool of IP addresses it can give out and I have the scope disabled. We're just going to go ahead and activate the scope. Now the DHCP server can respond to the client. Let's go back to our client and take a look.

I'm back on my client. Let's try our IP config. I still got my APIPA address. What you can do as an ipconfig /release, it is going to say I have nothing to release I don't have an address. Then I'm going to try an ipconfig /renew,

which tells it and give DHCP ago. There you can see that it's actually picked up an IP address from my DHCP server. We're going to go ahead and do an ipconfig /all.

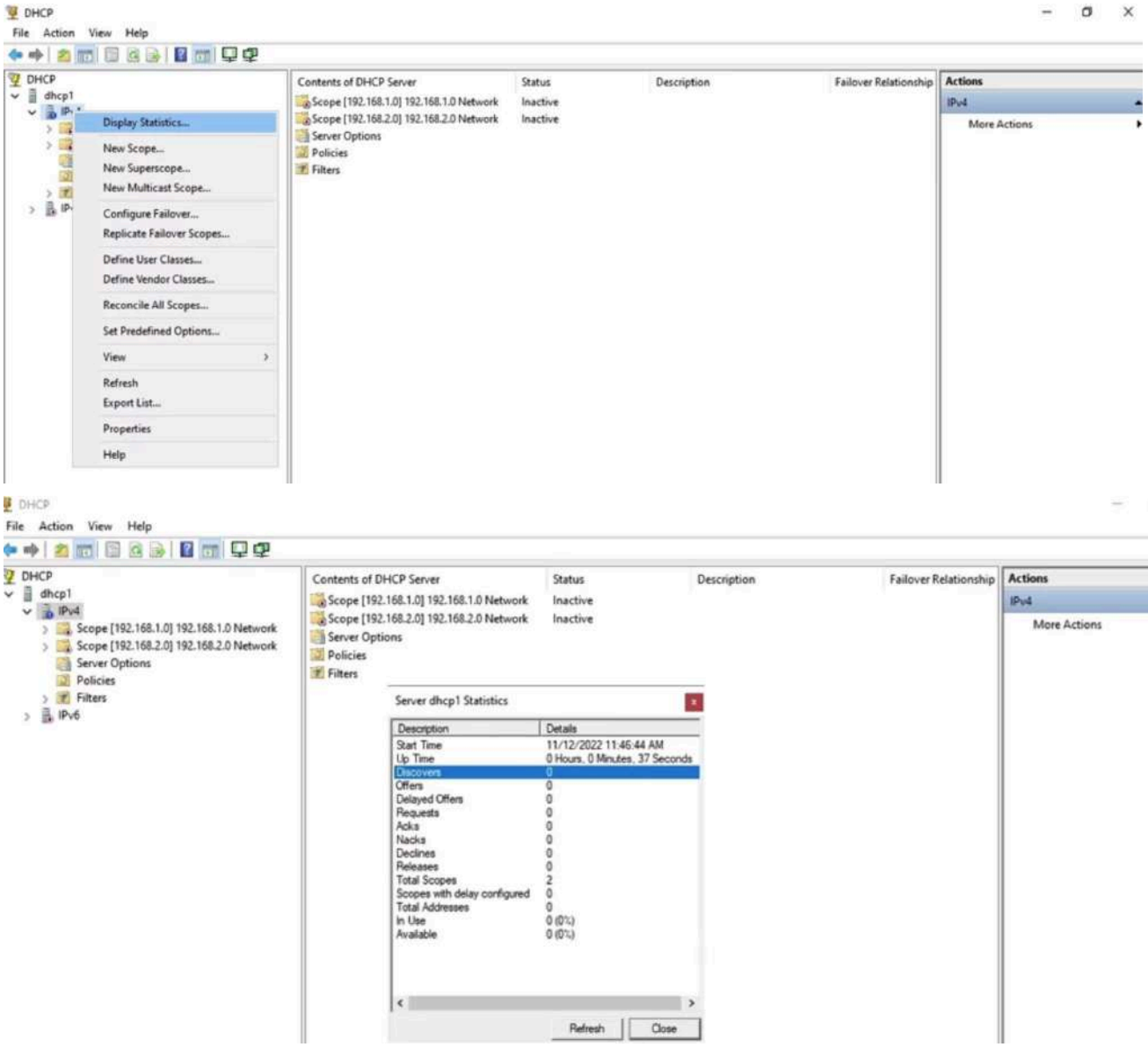


You can see, it gives us the date the lease was obtained and the date the lease expires. I don't know. I'm not great with math. I think it's about 24 hours, something like that. That's the lease. The client actually tries at 50 percent of the lease to renew it. Then if the DHCP server doesn't respond, it actually tries again at 87.5. I don't know who came up with these percentages, but anyway, then if the DHCP server still didn't respond, it would use it all the way up until the lease expires, and then it would just go back as if it had never had an IP address. In this video, we talked about static IP addresses, which are the ones typed in by an administrator and we looked at dynamic IP addresses, which are the ones obtained through DHCP. They come with a lease, which is the amount of time the client can use the address. If DHCP doesn't respond, the client will set in an APIPA address, which is an address on the 169.254.0.0/16.



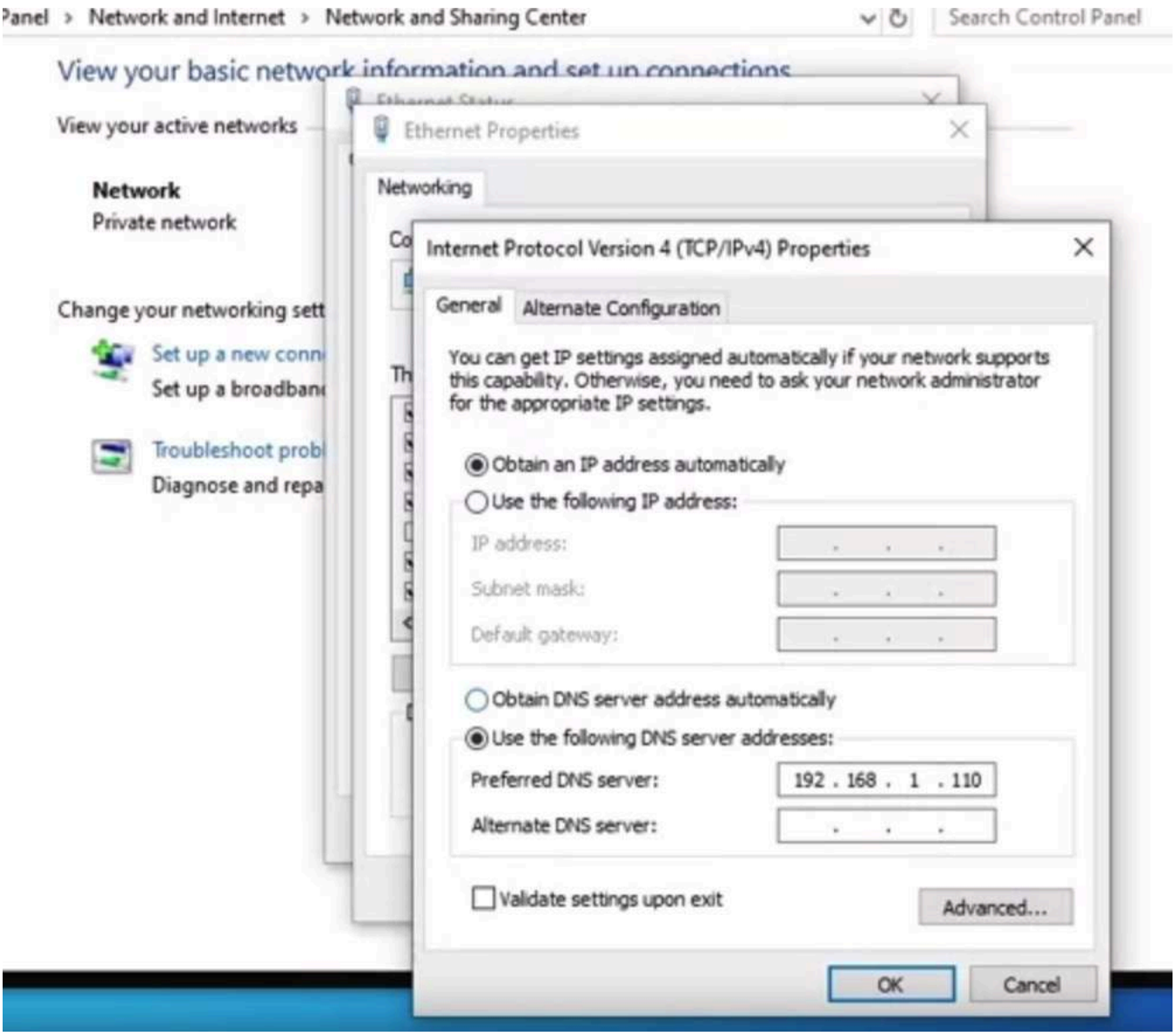
The DHCP Lease Process

We're going to take a look at the DHCP lease process. I'm actually here on my DHCP server and we're going to look at statistics because I think that's really helpful. That a little bigger. When a client set to obtain a dynamic IP address boots up, it sends out a broadcast to find a DHCP server. This broadcast is called a DHCP discover packet and if you look here on the screen where it says discovers, this is listing the discover packets that this DHCP server has received. Now, I just restarted DHCP so that all these counters would be zero and that's why it's there. The client must use broadcast because it has no IP address to use to receive replies. The broadcast will have the client's MAC address. When the server responds, it responds with a broadcast address to the MAC address of the client. We're going to go ahead and look at our client so that the DHCP and then come back and look at the discovers.



Our DHCP server is not going to respond to the client because I have two scopes here and they're both deactivated and I'm doing that for a reason. I just want to see the discovers right now. Let's go over to our client, set it to DHCP, and take a look at what happens. Here, my client, which is a Windows 10 machine, and we're going to go into the network card, change it over from a static IP address to use DHCP.

Perfect. Now it's looking for a DHCP server. Let's go back to the DHCP server and see what it looks like from that side. Back on our DHCP server let's go ahead and refresh, and you can see it got five DHCP discover packets.

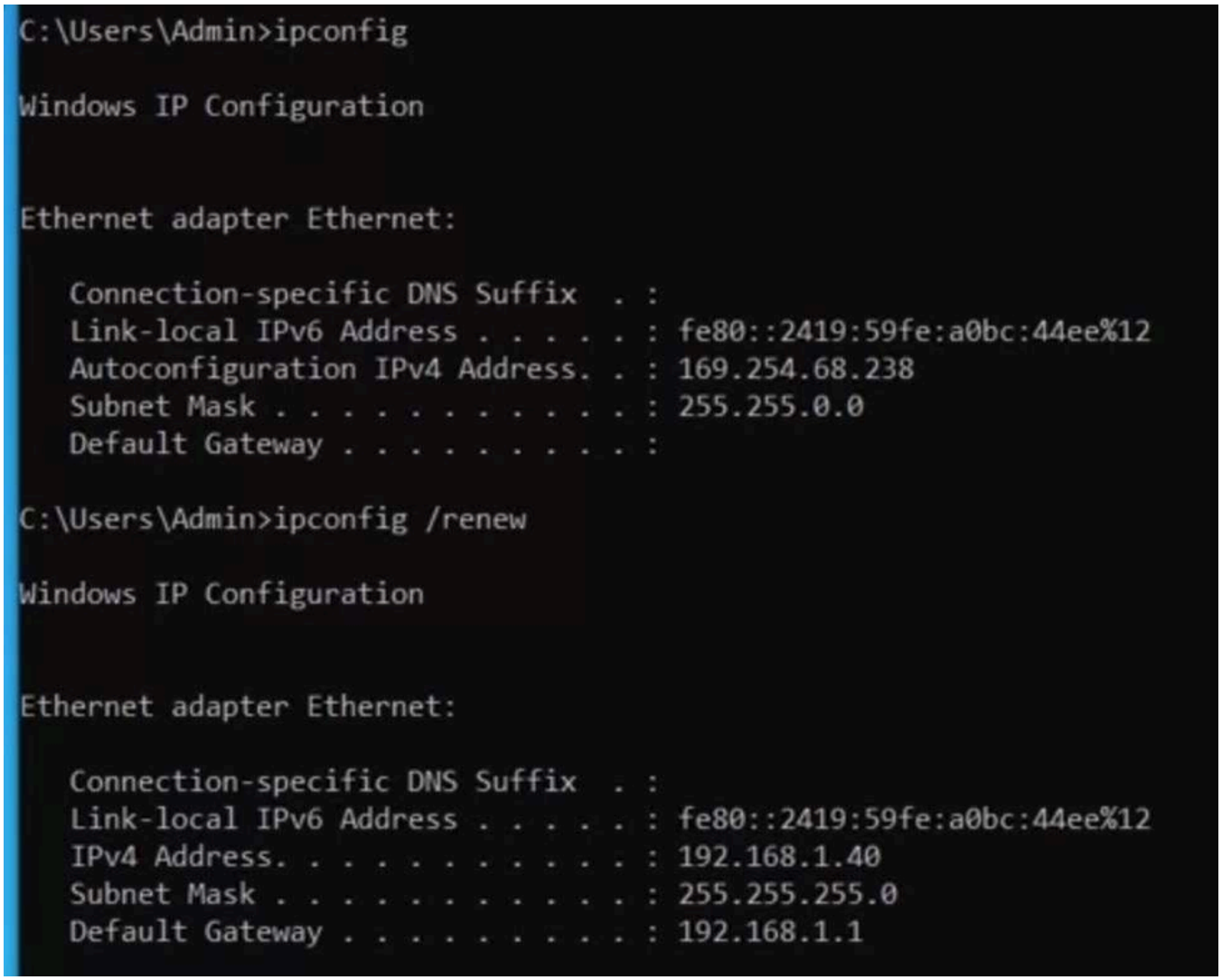


The client was looking for a DHCP server to help it out. This server didn't respond. Every DHCP server that gets that DHCP discover packet, what it would normally do is select an address from the pool of available addresses. Then it replies to the client with a broadcast called a DHCP Offer packet, which is this section right here for offers. The Offer packet would include a bunch of information like the IP address the server is offering, the subnet mask, the lease duration, or how long the client can use the IP address, the IP address of the DHCP server, and any additional information that's going with that address. The only time a DHCP server will not respond with a DHCP Offer packet is if the DHCP server doesn't have any available addresses or it can't support that host. In this case, our DHCP server doesn't have any available addresses because it doesn't have any addresses. That's why it's not responding. Let's go ahead and activate the scope. The client is going to continue trying for a certain amount of time, but after a while it will give up. You can see the statistics here just went a little bit crazy. I'm just going to reset the DHCP server so that all that will go to zero.

While it's doing that, we're going to pop over to the client and just tell it to try again in case it stop trying.

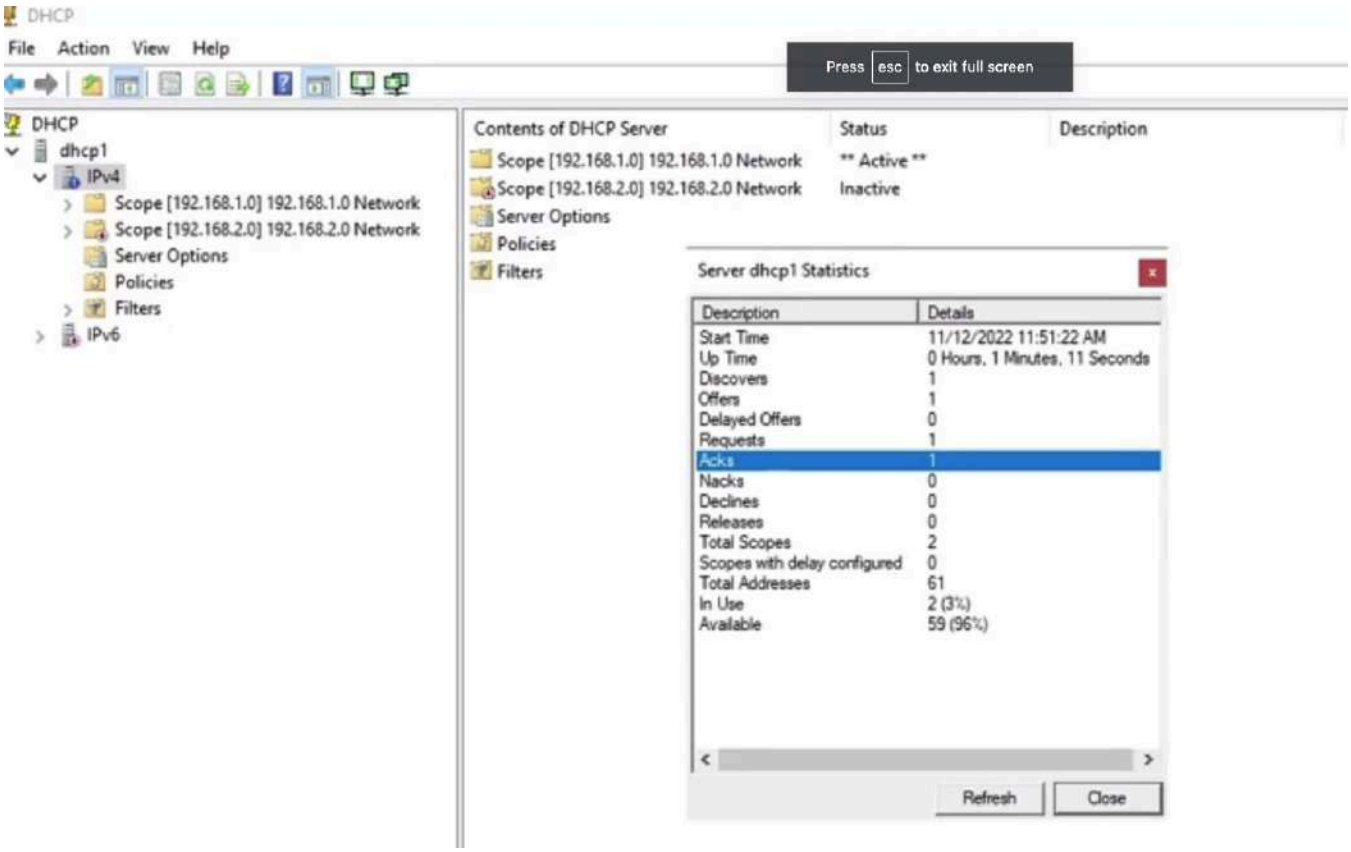
No discovers. I'm thinking the client gave up. Let's tell the clients try again. Back on our client. I'm going to use the command prompt. I'm just going to do an ipconfig.

Right now we have an APIPA address, so let's try a renew, and it picked up an address from the DHCP server.



Let's go back to the DHCP server and check out what happened.

We'll go into our statistics and you can see now it's really different. It got the discover, it made an offer to the client.



Now the client is going to get an offer from every DHCP server that gets the discover. If multiple DHCP servers respond, the client chooses the first response it gets and it broadcasts a DHCP request packet, naming the server it chose and asking to lease the offered IP address. You can see here, there's the DHCP request packet, the client saying, "I'll take you." This is the only DHCP server that responded. Any DHCP servers that offered, but we're not chosen, put the offered address back in the pool. The selected DHCP server marks the addresses leased and sends the client a DHCP ACK packet confirming the client has successfully leased the address. There's the ACK packet that went from this DHCP server back to the client. You've got that address. Go ahead. You do not need to memorize the names of the packets. However, there's two things you should take note of.

First, this is all done with broadcasts and this can present a problem if you have multiple networks, but want to use only one DHCP server. Since routers do not pass broadcast traffic by default, only the clients on the same network as the DHCP server will successfully get addresses.

The second thing is the client picks the first DHCP server that responds. If there's an unauthorized DHCP server, and in technology when you have unauthorized hardware, we say rogue. There's a rogue DHCP server and the client gets the response from the rogue DHCP server. First, it's going to pick the rogue. Typically, if I come in here and there's lots of vendors for DHCP, I'm just showing you Microsoft because they have this nice statistics. But if I'm working with Microsoft DHCP, I can tell you what the problem is just by looking at the statistics. The clients are trying to get an IP address from DHCP and discovers are not going up. That indicates that there's a hardware problem. There's just not physically a connection between the client and the server. If discovers are going up, but offers are not going up, as we saw at the beginning of

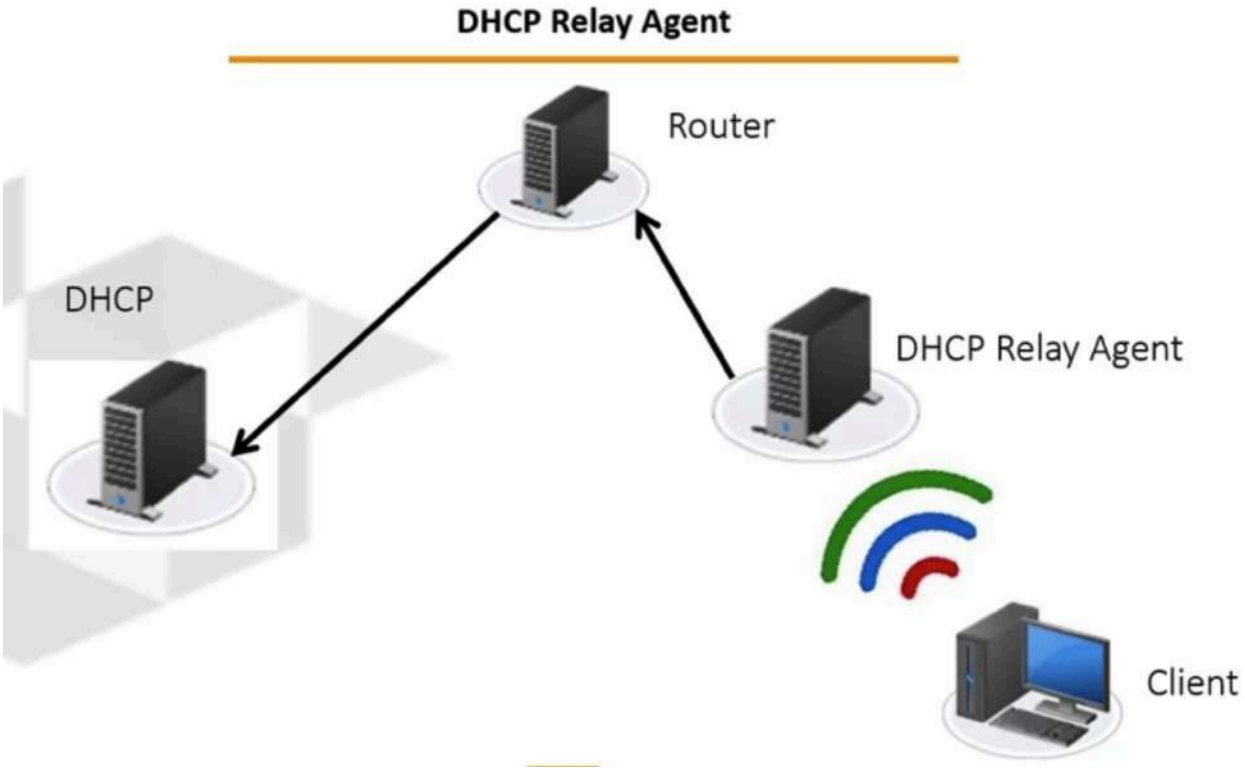
this video, it means that the server thinks it can't help the client so maybe it's out of IP addresses or the IP address on the server is set wrong but it's something on the DHCP server itself.

If discovers are going up, offers are going up, but requests are not going up, that tells me that there is a rogue DHCP server in the environment. If the client is not choosing this DHCP server, why would that be? Because it chose a different DHCP server. I have never seen it where discoveries are going up, offers are going up, requests are going up, but ACKs are not going up. I don't know what that would be. That's it for this video. In this video, we looked at the DHCP lease process. We explored the broadcast that the client send out to obtain an IP address from the DHCP server and we talked about some of the difficulties that could come from that process.

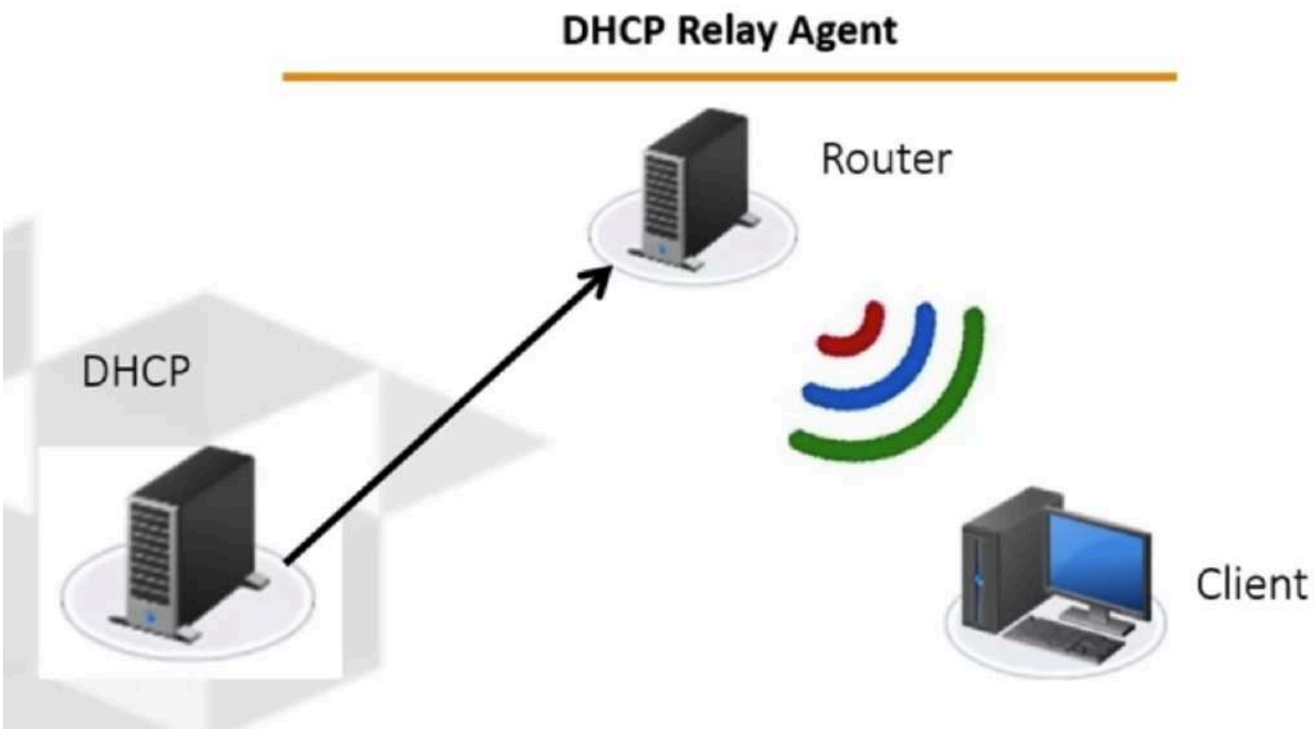
Centralized DHCP

we're going to talk about centralized DHCP. If you want to use centralized DHCP, meaning one DHCP server for multiple networks, you will need to do something for the clients that are not on the same network as the DHCP server because remember the clients use broadcasts, and routers don't pass broadcast traffic. There are two options, a DHCP relay agent or an RFC 1542 compliant router. Let's take a look at this scenario. I have one DHCP server on the left, which is on one network, and then on the right I have a client that's on a different network. To enable this client to receive an address via DHCP, I need a DHCP relay agent. What happens with the relay agent is that the client sends out the broadcast hope is their DHCP server in the house, the relay agent listens for DHCP broadcasts and then relays them directly to the DHCP server.

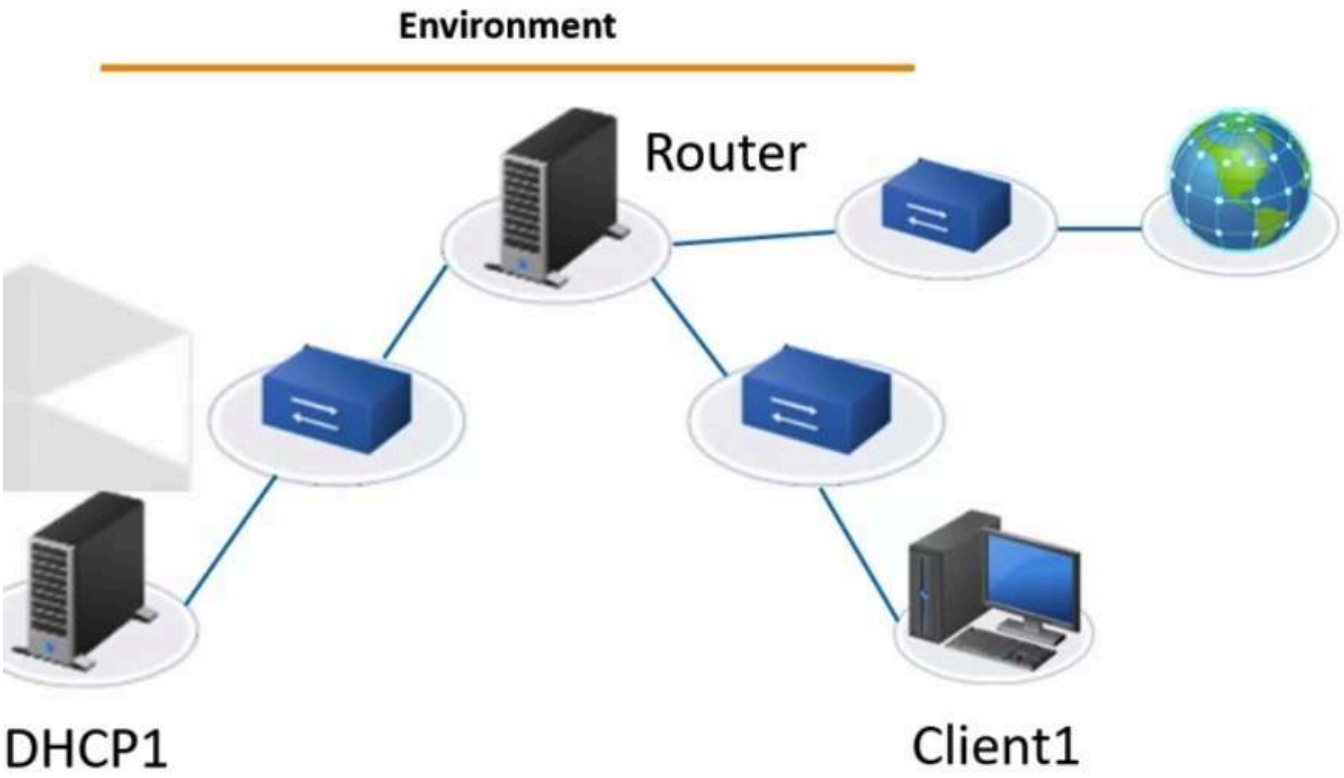
When the DHCP server replies, the DHCP relay agent broadcasts the reply on the network so the client can get it.



With an RFC 1542 compliant router and usually that would be a physical router, then it's the same thing but effectively what's happening is that the router is configured to pass the DHCP broadcast to a specific DHCP server on another Nick on the router. RFC 1542 it's just the standard for routers, so if it meets that standard, you can program it to do this.
Here it'd be a physical router, the client sends out the broadcast, the router relays that to the DHCP server, when the DHCP server replies, the router is going to put that out as a broadcast on the network for the client to get.

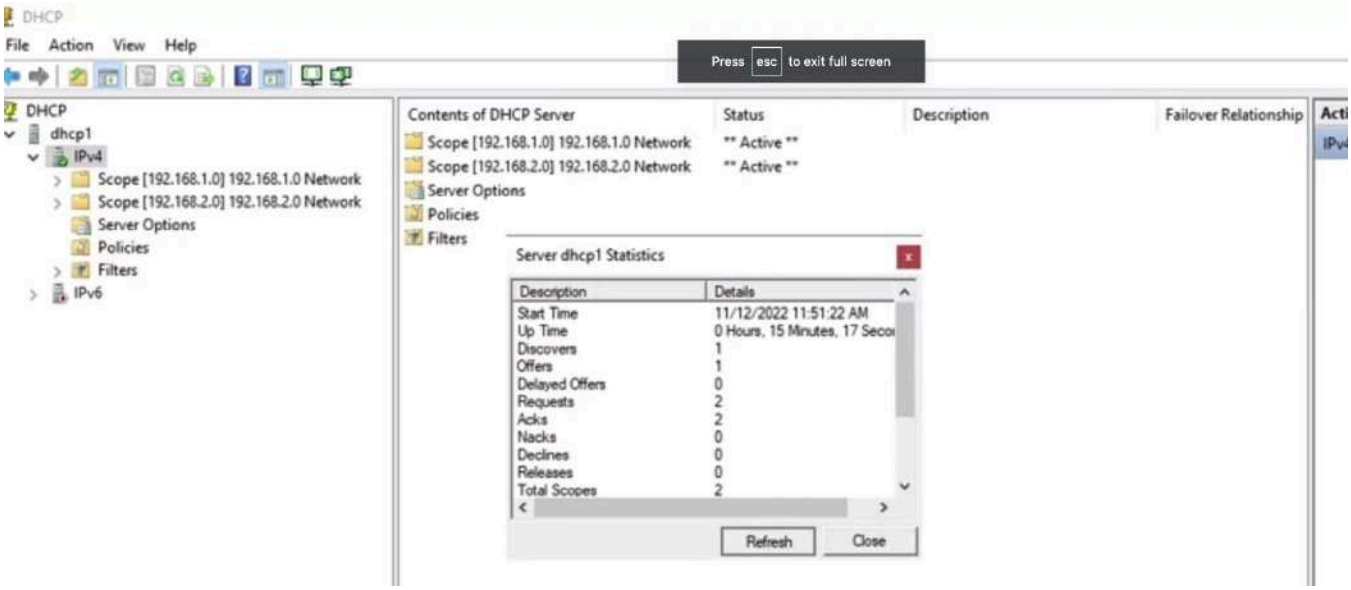


We're actually going to check this out in the software but before we do, let me show you what the environment looks like. Right now I have a router. It's actually made with a Microsoft server, so it's not an RFC 1542 compliant router. We're going to be turning it into a DHCP relay agent, so combining the two. That router is attached to three switches. One switch on LAN 1 has a DHCP server connected to the switch. Second switch, LAN 2 has a Client 1 connected to the switch. Then the third switch is connected to the Internet.

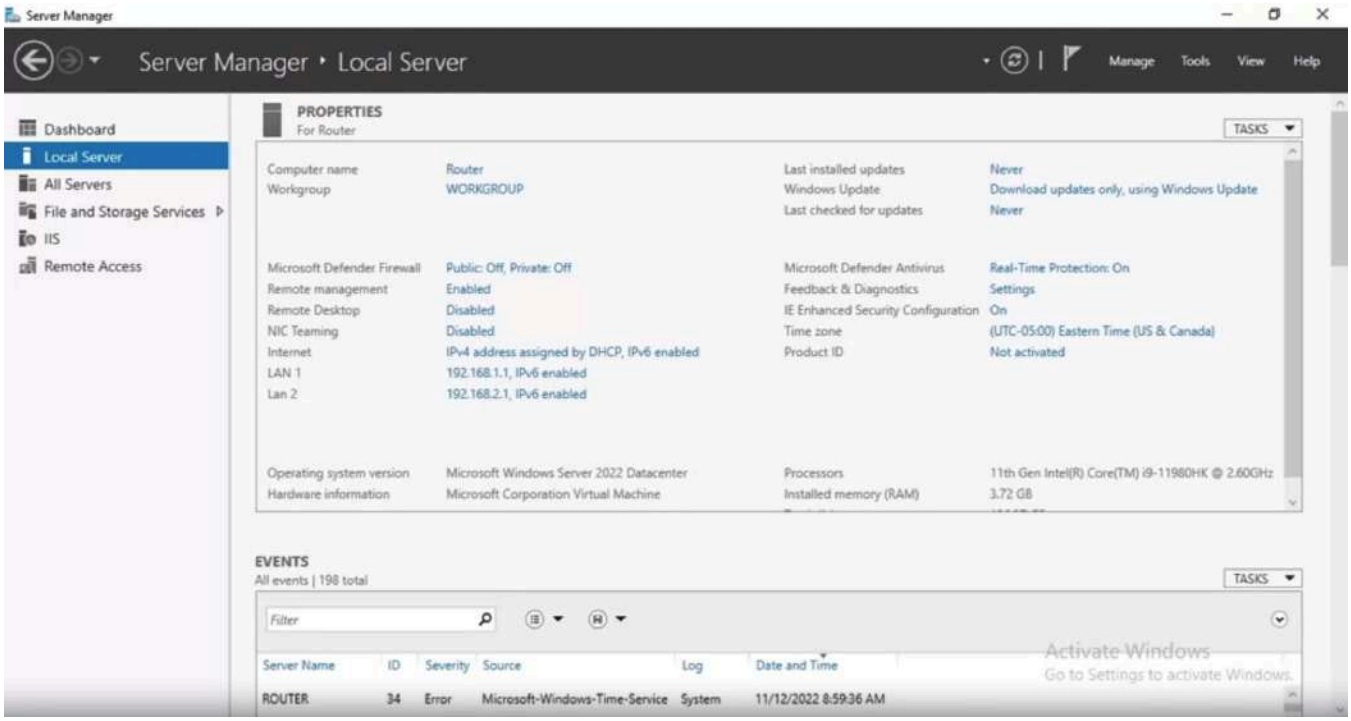


Let's go take a look at our DHCP server. We're here on the DHCP server, LAN 1 is this 192.168.1.0 network, and that's actually the network that this particular server is on. You can see its name is DHCP1, and there's its IP address 192.168.1.110. LAN 2 is 192.168.2.0 network, and the client is actually attached to that. Right now let's go in and look at the statistics. So far it's seeing one discoveries made an offer, a couple of requests and acknowledgments.

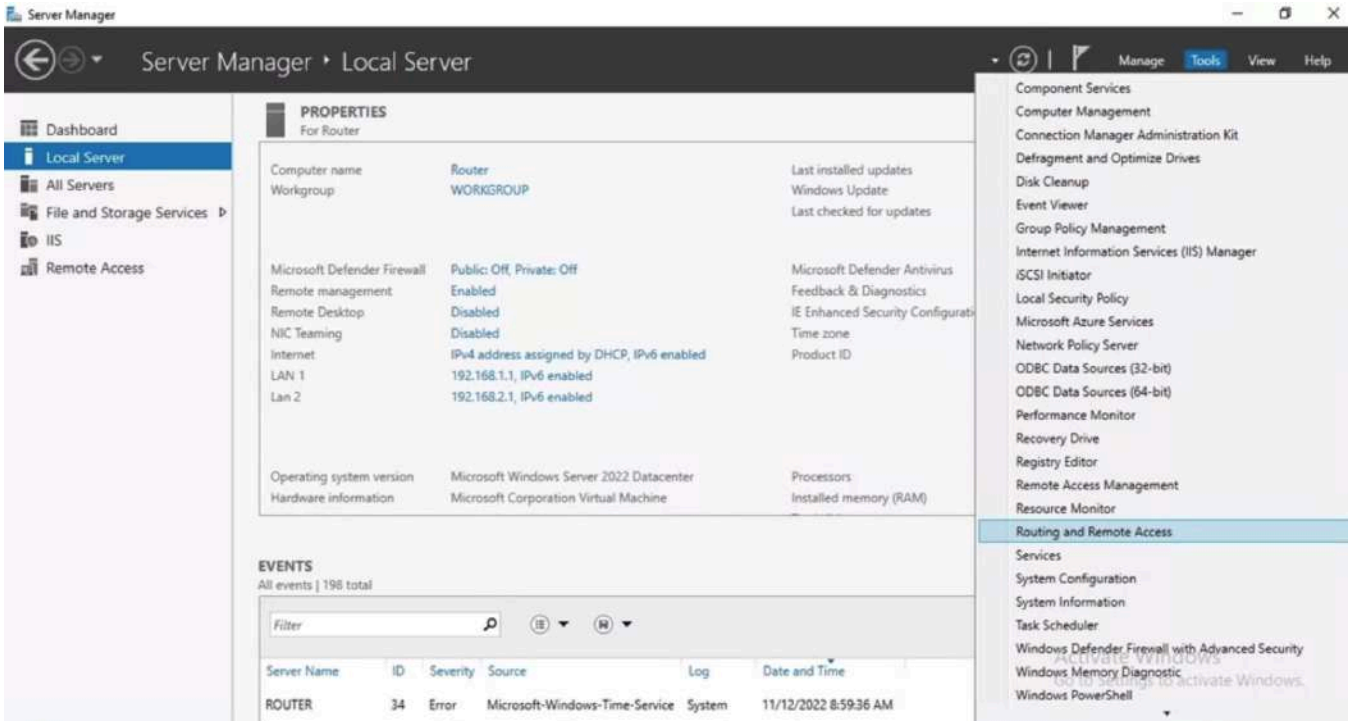
Let's go over to the client and see what happens over there. I'm here in my client. It currently has a static IP address for the 192.168.2.0 network, and it's using when I do 192.168.2.1 is the gateway, and it has Internet access, no problem, so everything is working great. Let's change this over to DHCP. I'm going to go into my network card and say no, you don't need a static IP address. Let's do DHCP instead and you can see it hasn't picked up an IP address. Some point it's going to fail over to IPv. There's the IPv address, and the problem is of course, the client is sending out their broadcast but the router is not passing that broadcast over to the DHCP server. Let's just pop over to our DHCP server and confirm that. Back on the DHCP server, I'm just going to refresh my statistics, and you can see DHCP discovers is still one. The client never got that broadcast.



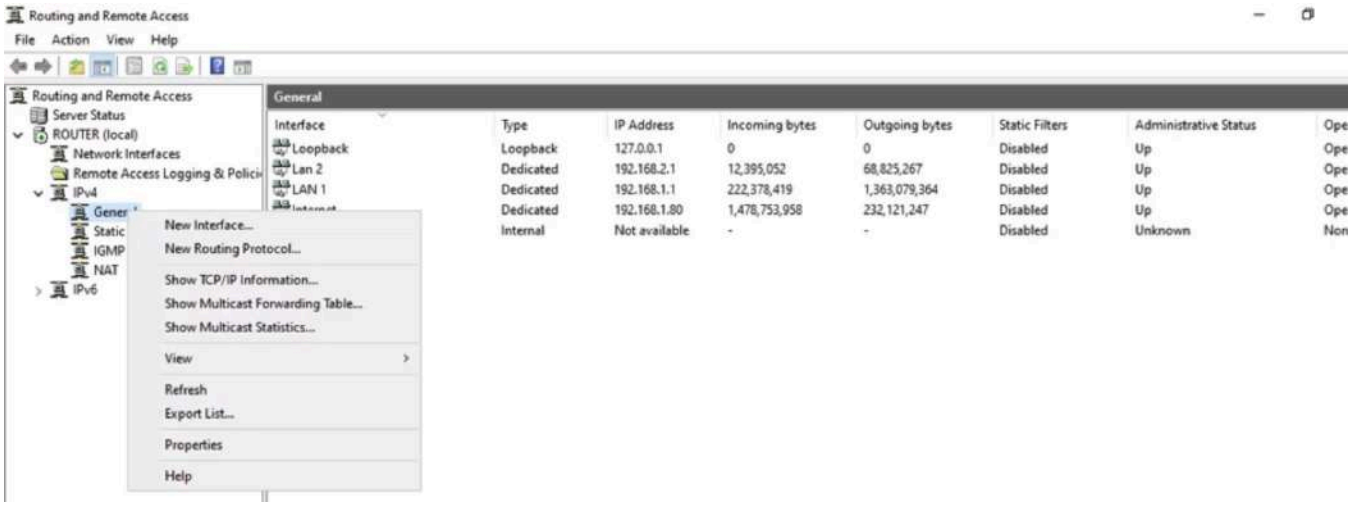
What we need is a DHCP relay agent, so we're going to go over to our router and configure a DHCP relay agent. I'm here on my router, which I have creatively named router. You can see it's got two IP addresses; 192.168.1.0, 192.168.2.1, and in a Microsoft server the software that allows it to be a router is routing and remote access. Right now it's running NAT, and it's doing general routing. We're going to add in a DHCP relay agent.

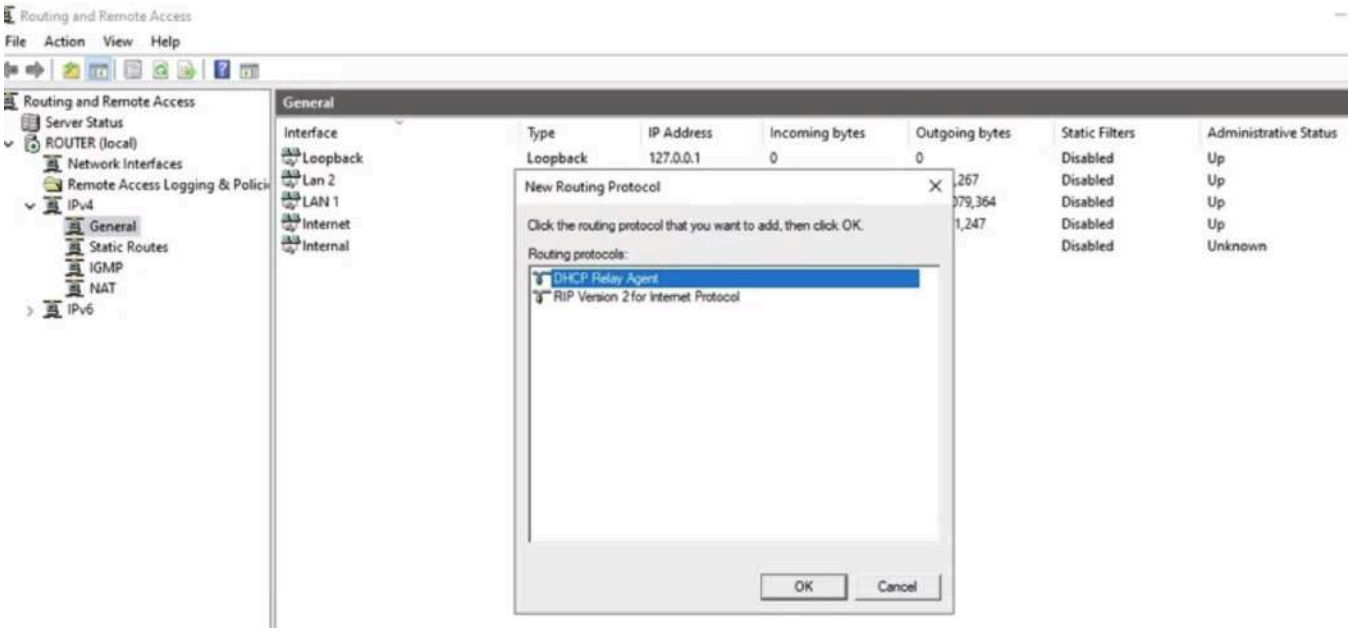


so we're going to go over to our router and configure a DHCP relay agent. I'm here on my router, which I have creatively named router. You can see it's got two IP addresses; 192.168.1.0, 192.168.2.1, and in a Microsoft server the software that allows it to be a router is routing and remote access. Right now it's running NAT, and it's doing general routing. We're going to add in a DHCP relay agent.

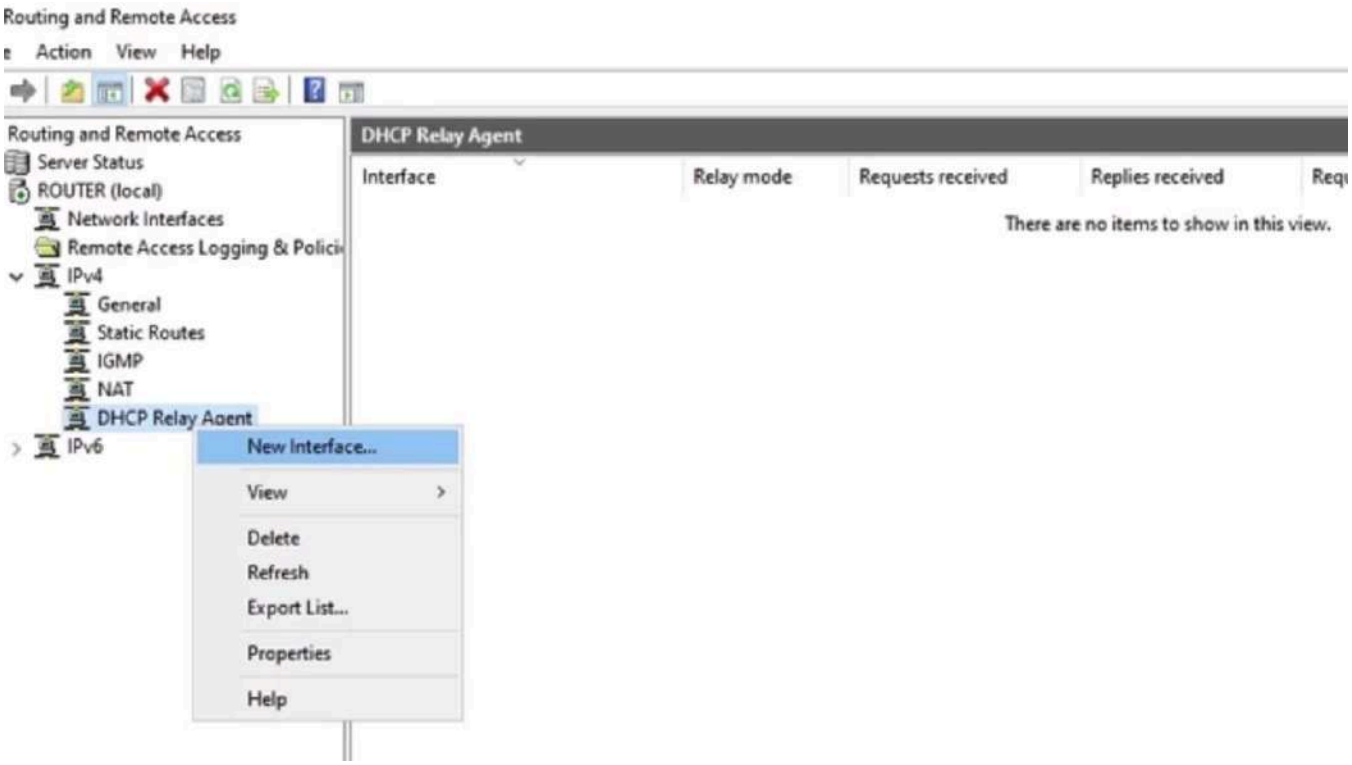


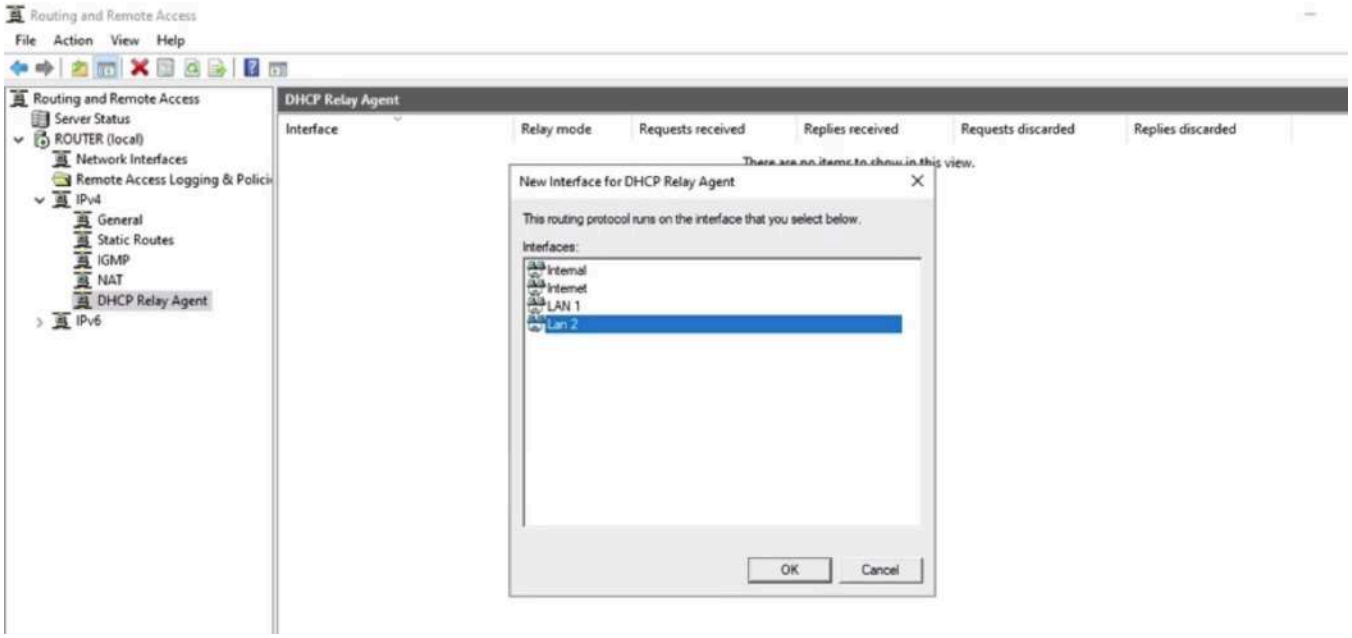
I'm going to do a new routing protocol. There's my DHCP relay agent, and it's just that easy.



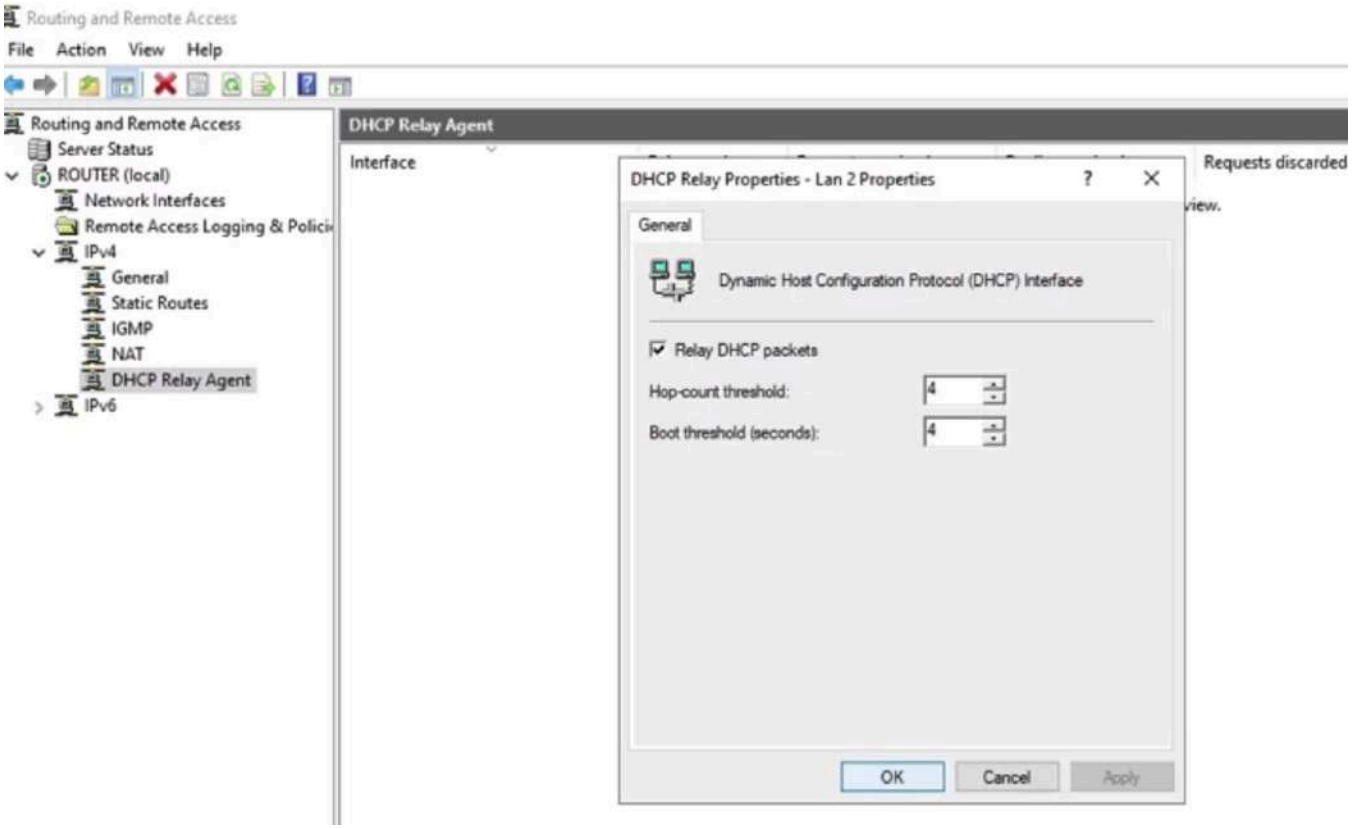


Now two things we have to tell the relay agent. The first thing is, what network card should it listen for broadcasts on? I'm going to go ahead and add a new interface and say, you should listen on LAN 2. That's the LAN that doesn't have a DHCP server, and yes, you should relay DHCP packets from LAN 2.

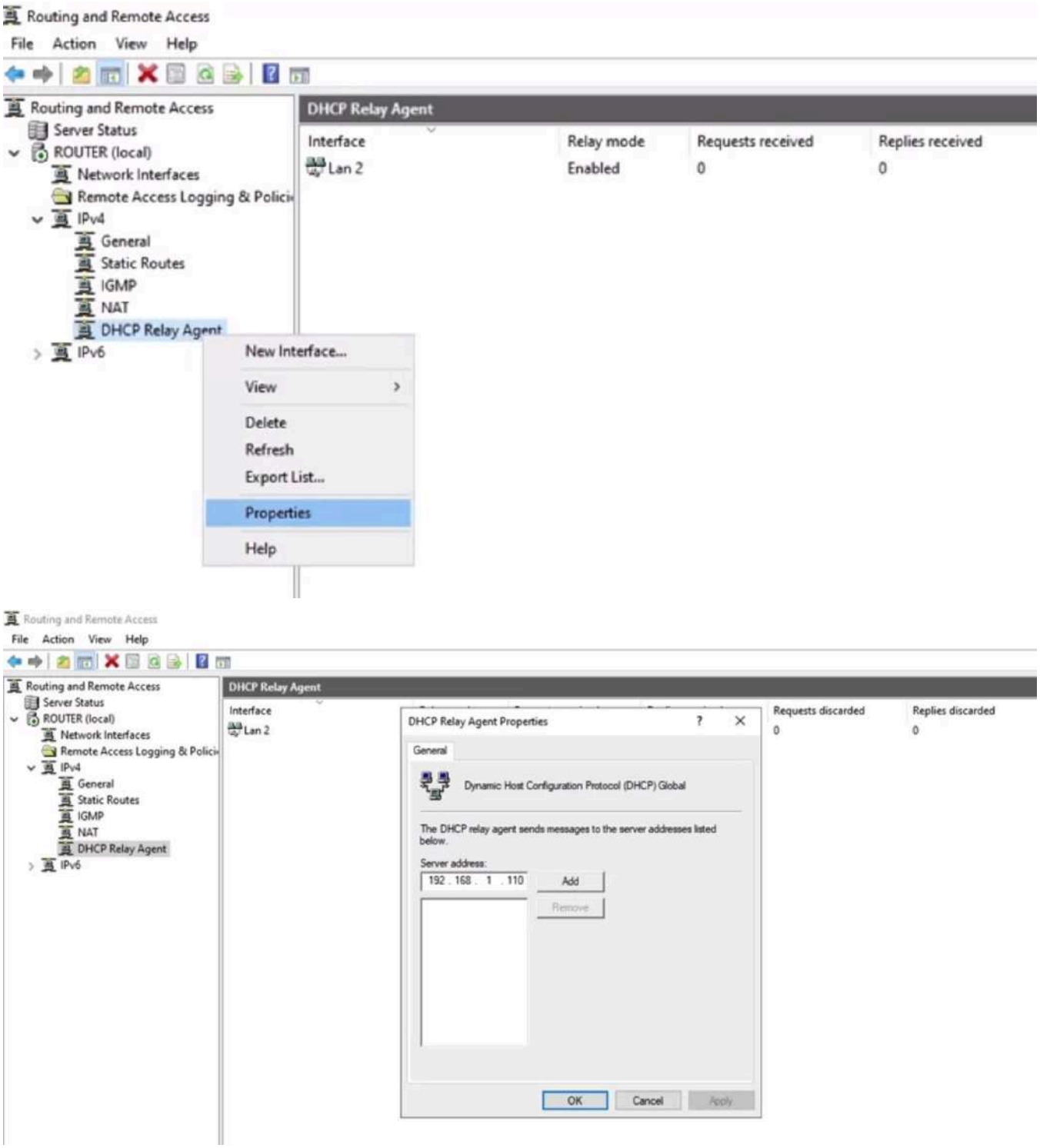




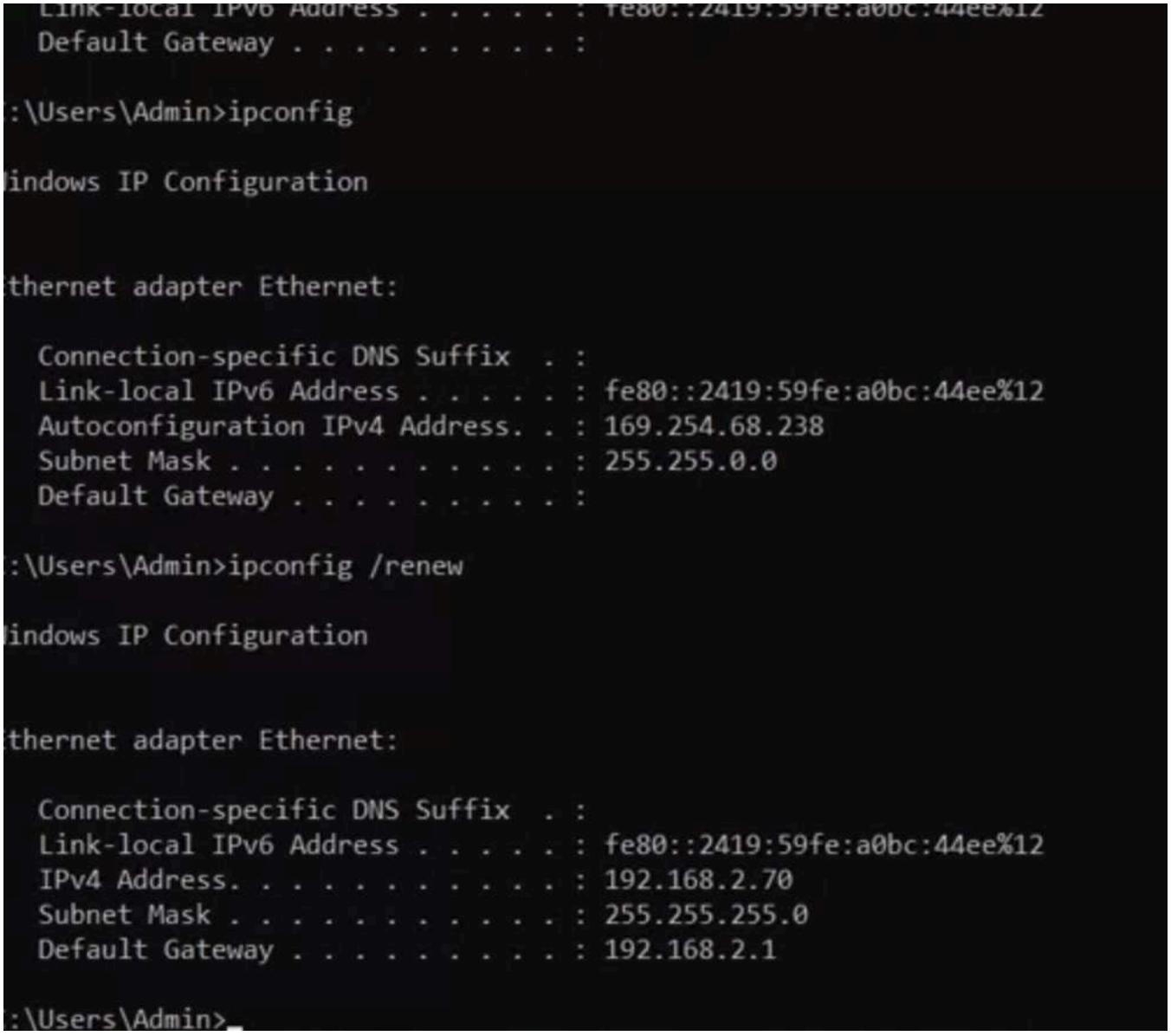
.



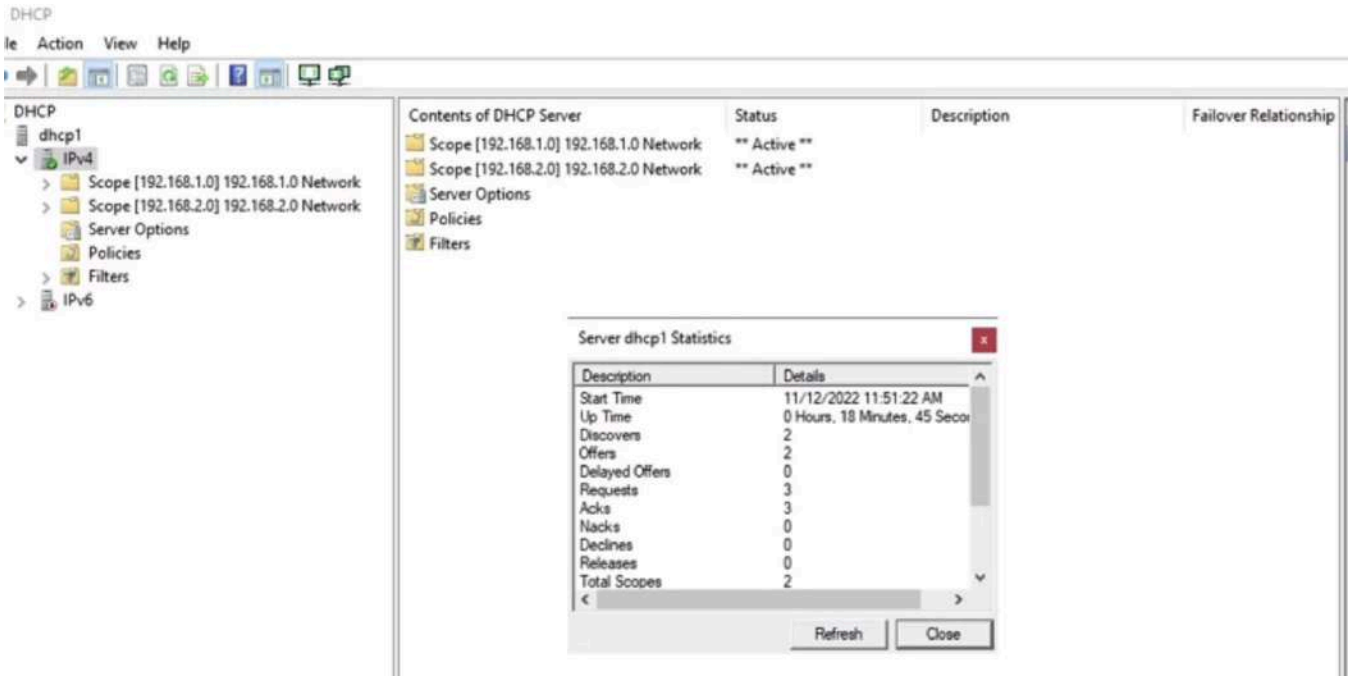
Second thing my DHCP relay agent is going to need to know is, where do you send those packets that needs the IP address of the DHCP server? We would go to Properties and say, well, you can send it to 192.168.1.110



Now my relay agent knows everything it needs to relay those packets. Let's go back to the client and tell it to try again. Back on our client, let's try an ipconfig/renew. It takes a little bit more because it's going through the relay agent to the DHCP server and then back through the relay agent but there's our address, 192.168.2.70. It looks great.

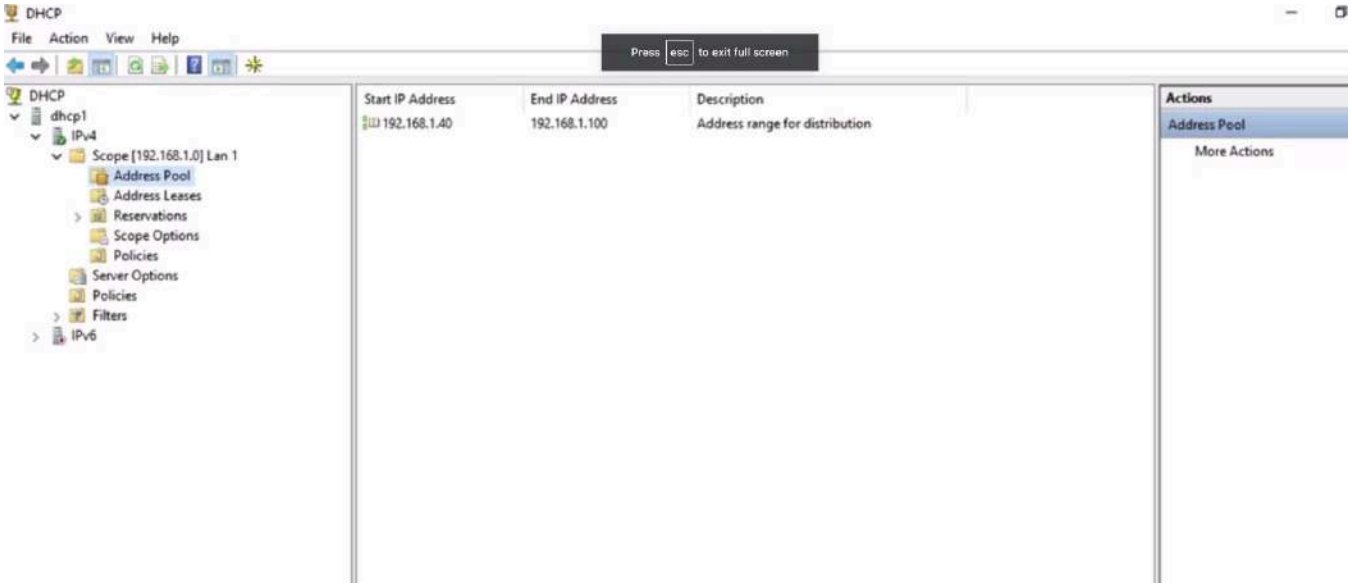


Just to confirm, it will take one last look at the DHCP server. Let's go ahead and open our statistics, and you can see that the discovers used to be one now it's two, so it did in fact get that discover via the DHCP relay agent. That's it for centralized DHCP. We took a look at the issue with centralized DHCP, which is that we have to have some way to get the broadcasts from the network that the clients are on to the network with the DHCP server, and there's two ways we can do that. One is with a DHCP relay agent, which is a server that's configured to relay the packets, and the other would be with an RFC 1542 compliant router.

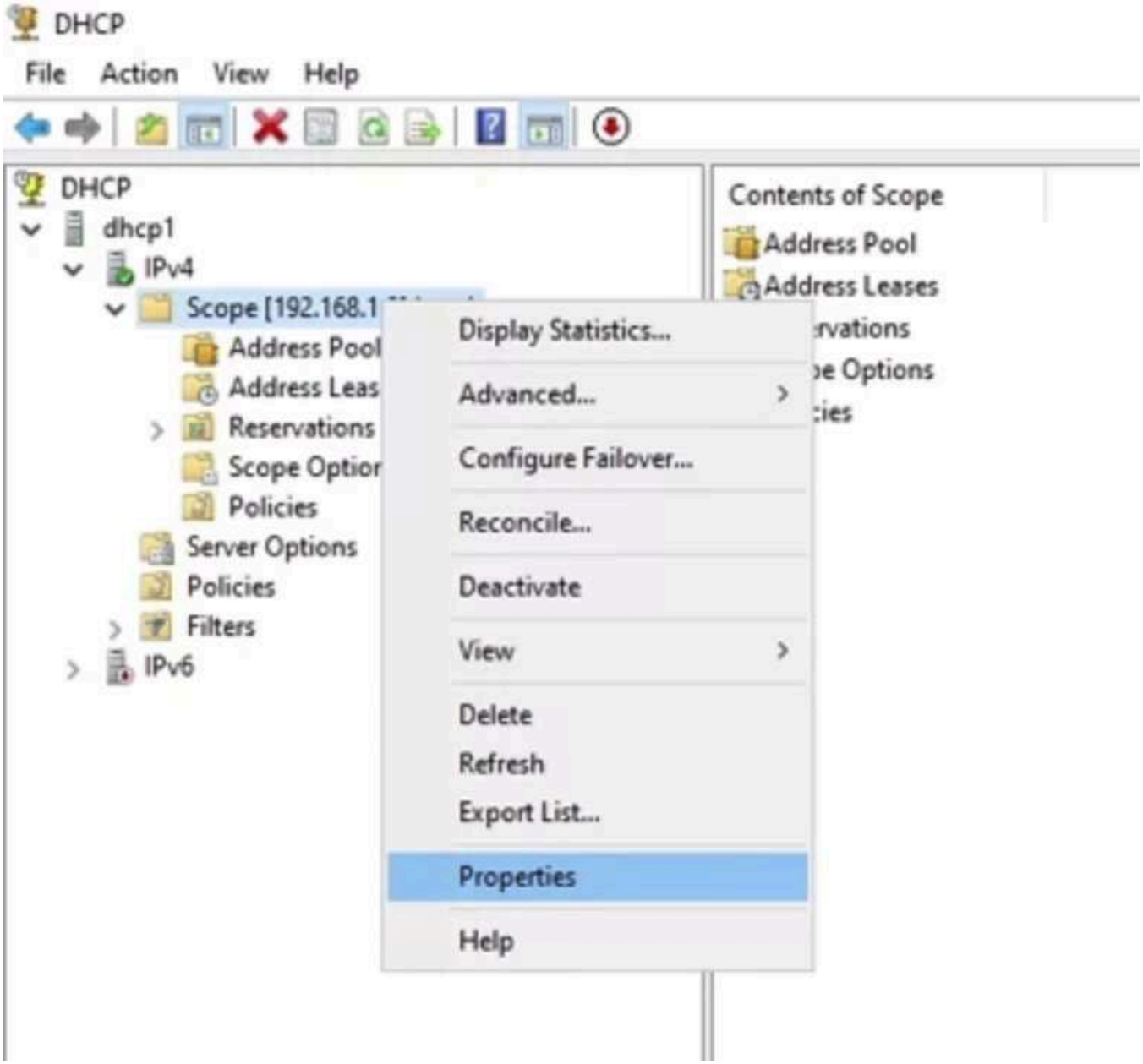


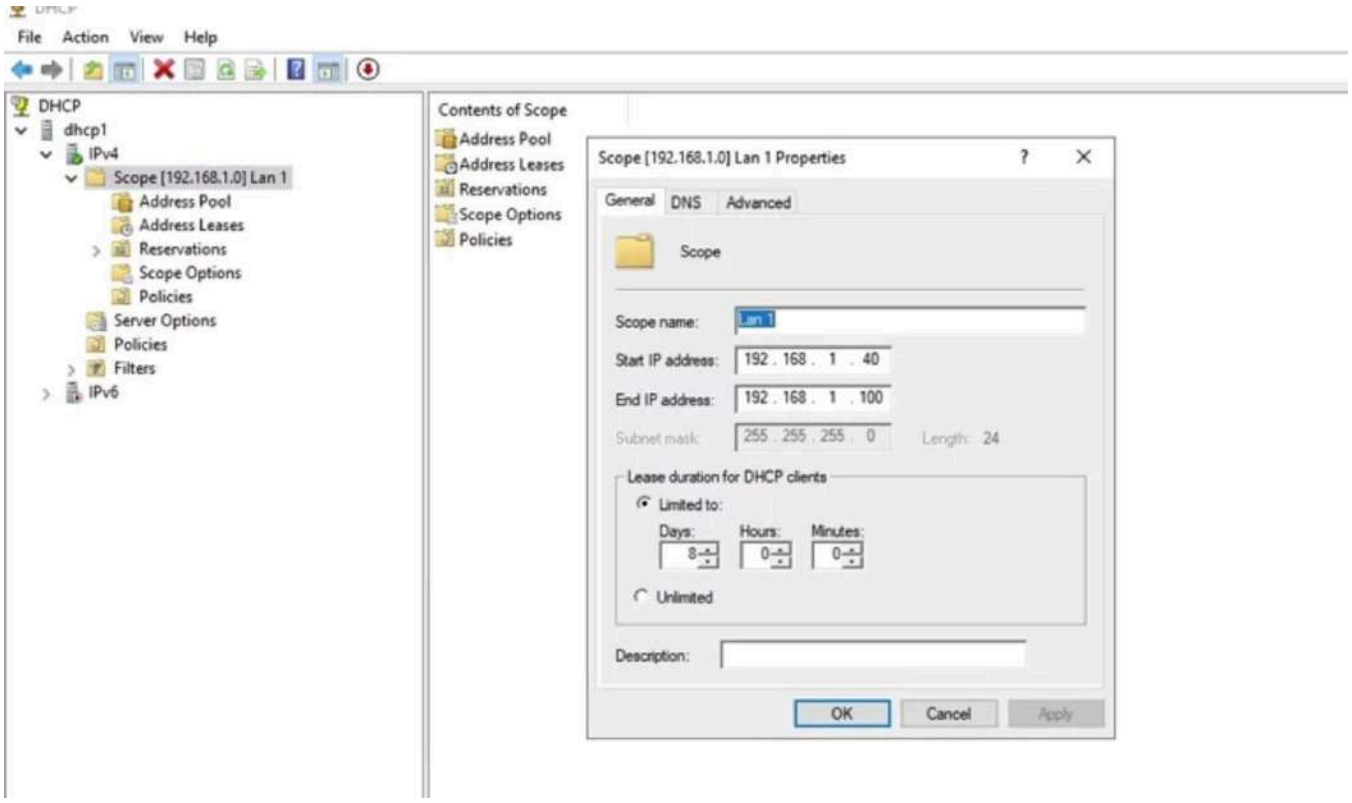
DHCP Server Settings

We're going to take a look at some of the DHCP server settings. I'm here on my DHCP server and the pool of addresses the DHCP server can give out is called the scope. You can see here I have a scope named Lan 1 on the 192.168.1.0 network. It has an address pool that runs from 192.168.1.40 to 192.168.1.100. This scope includes the addresses that can be issued to clients along with the subnet mask.

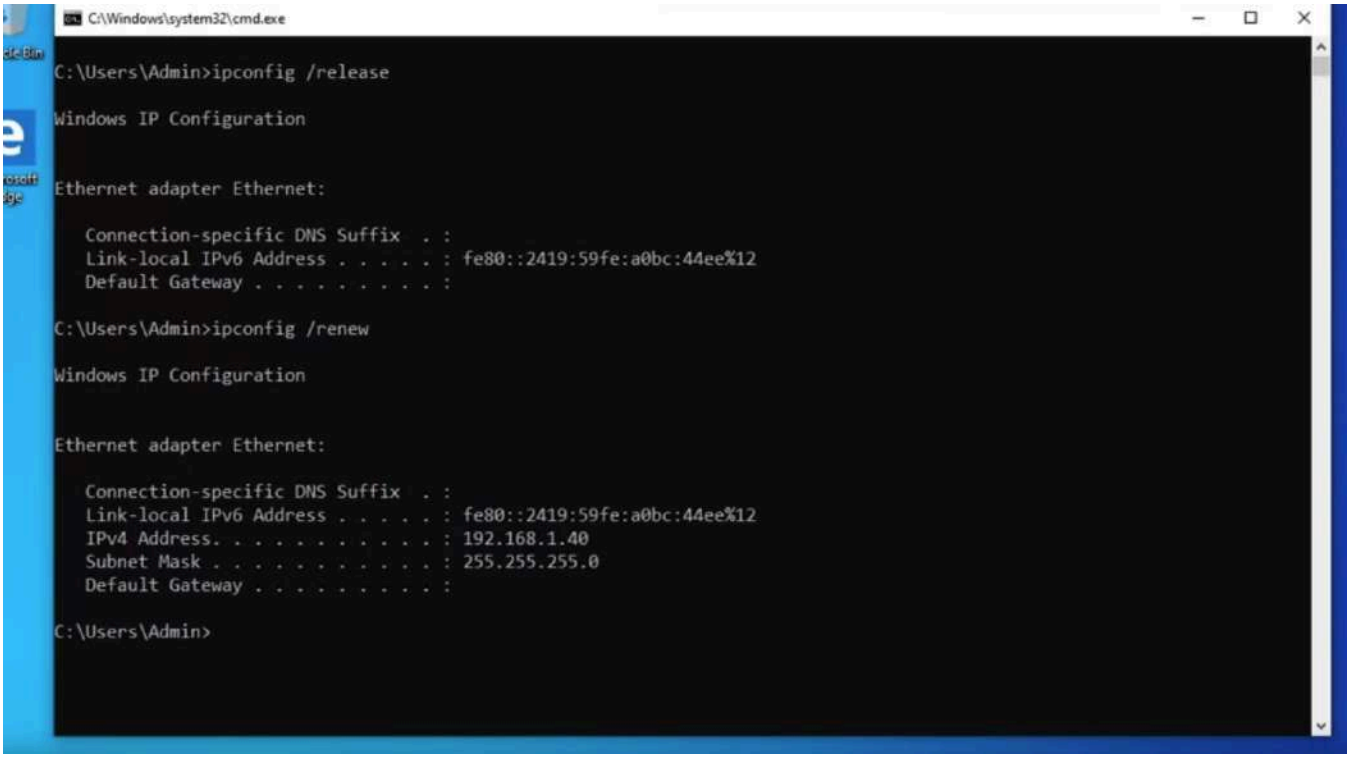


If I go to the properties here, you can see I can adjust the addresses. On a Microsoft DHCP server, you can't adjust the subnet mask. Every vendor's DHCP works a little bit differently, but they're all very similar.



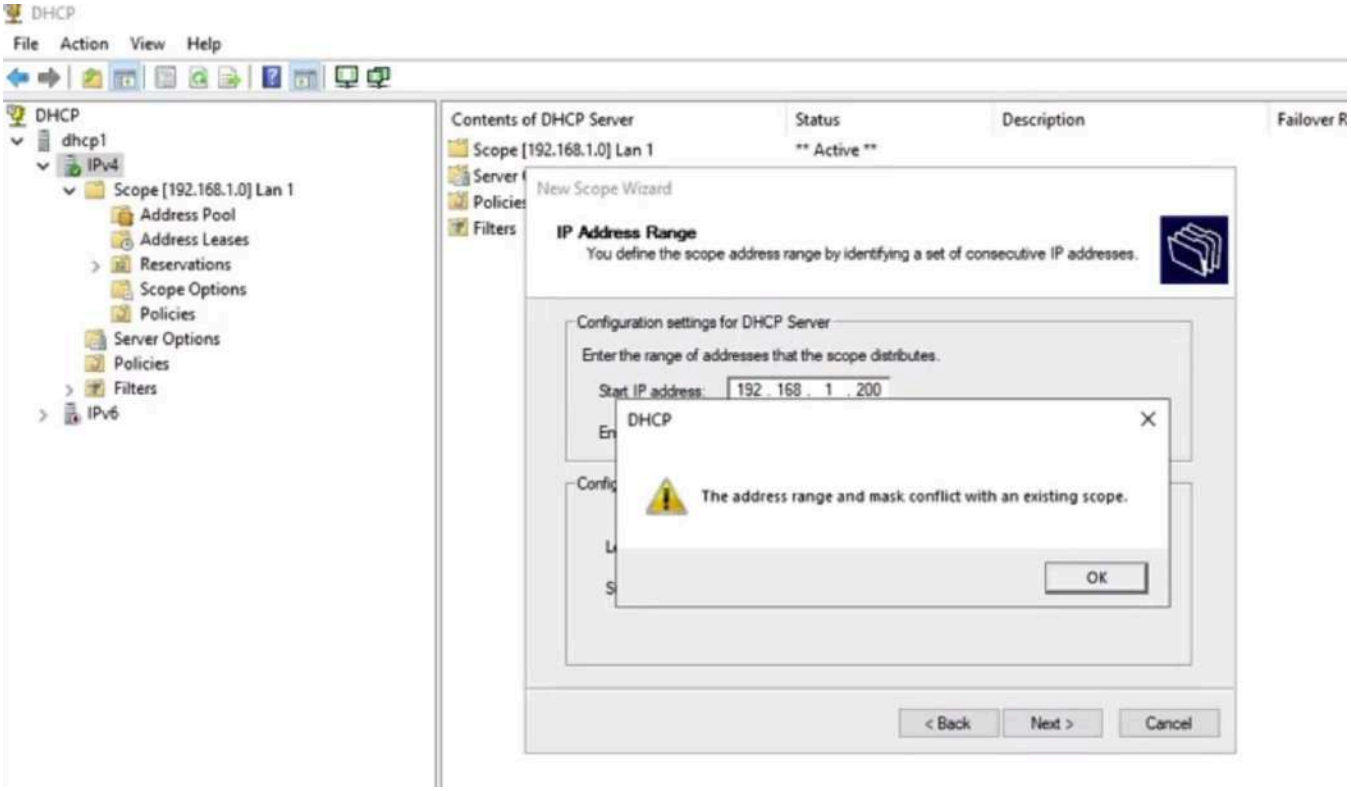


Let's go over to our client and obtain an IP address from this scope. I'm just going to go in and do an ipconfig /release because it already has an IP address. Now it doesn't have one. We'll do our ipconfig /renew to ask it to obtain a new address. There you can see 192.168.1.40. The DHCP server just gave it the very first address that's available.

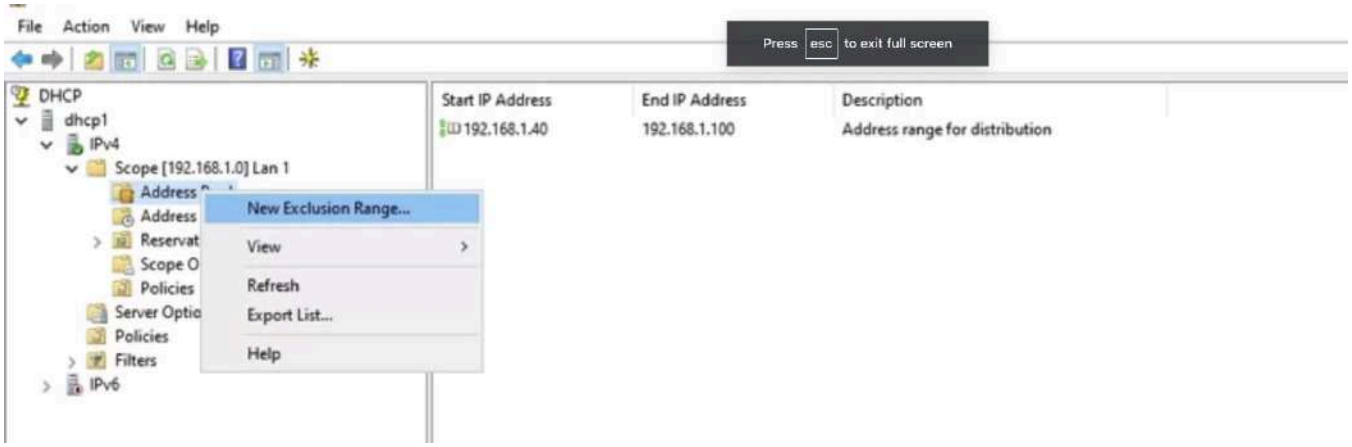


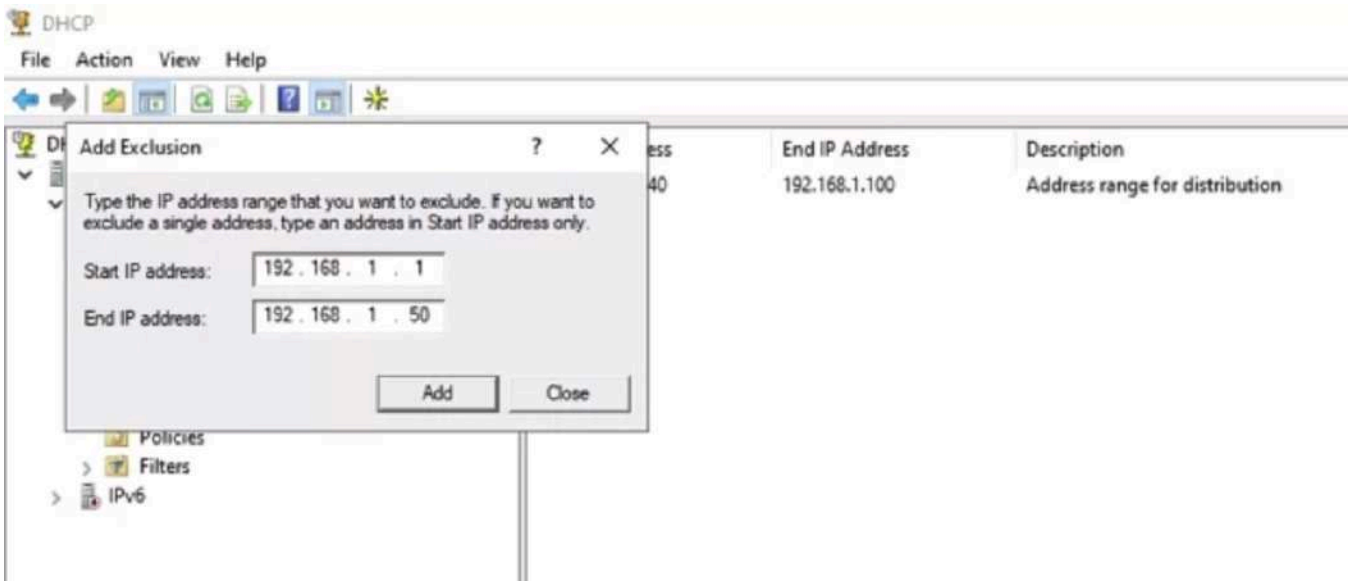
Let's go back to our DHCP server. For most DHCP servers, you can have just one scope for that particular network. This scope is giving addresses from 40 through 100. What if I also wanted to give out addresses from, let's say 200 to 250. If I go in and I try to make another new scope,

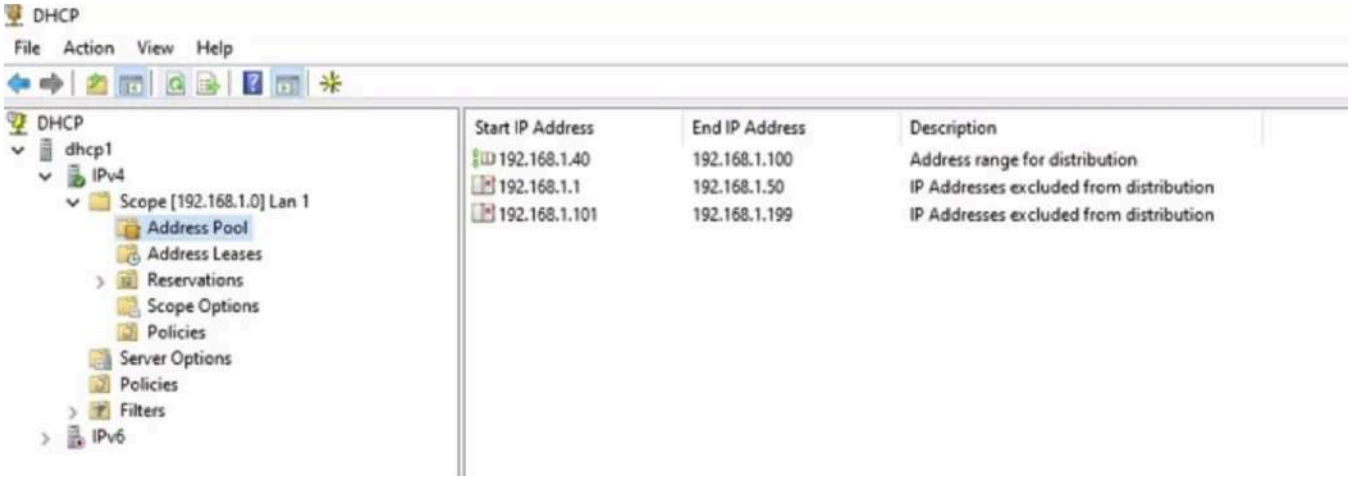
I'd like to give out from 200 through 250, it says no, you can't do that, you already have a scope. How would I do that? I would do that using exclusions.



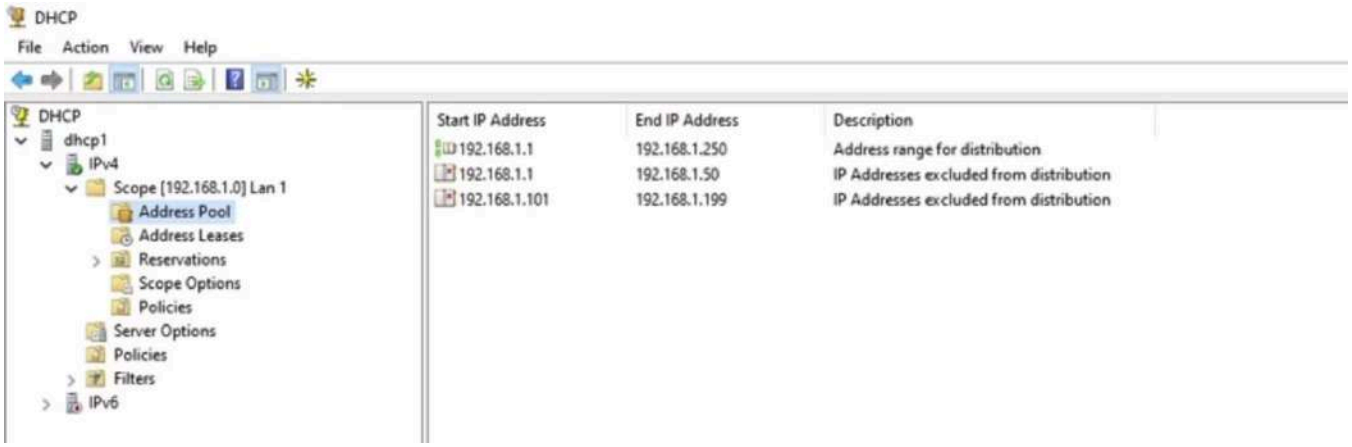
Exclusions are addresses that should not be given out to clients. But we don't usually use them because we're crazy and we just want to hand out all these weird addresses. Usually, they're used to identify static addresses in the network. We're looking at this scope, we can probably intuitively say, well, maybe 1 through 39 isn't used as static addresses in this environment. But a really easy way to document this is to have the address pool run the entire range of the network and then provide exclusions which would tell anybody looking at the DHCP address pool, those are static addresses. We're actually going to go ahead and change our address pool. We'll have it run from 1 to, let's say 250. Now we'll add some exclusion. Let's take out the first 50 addresses as static addresses. Our first exclusion range will run from 192.168.1.1 to 192.168.1.50. That will give out from 51-100. If you're looking at the background of the screen and, you say, well, it still looks like it runs from 40-100, it's just because I haven't refreshed the screen. Let's just do another exclusion from 101 through 199.



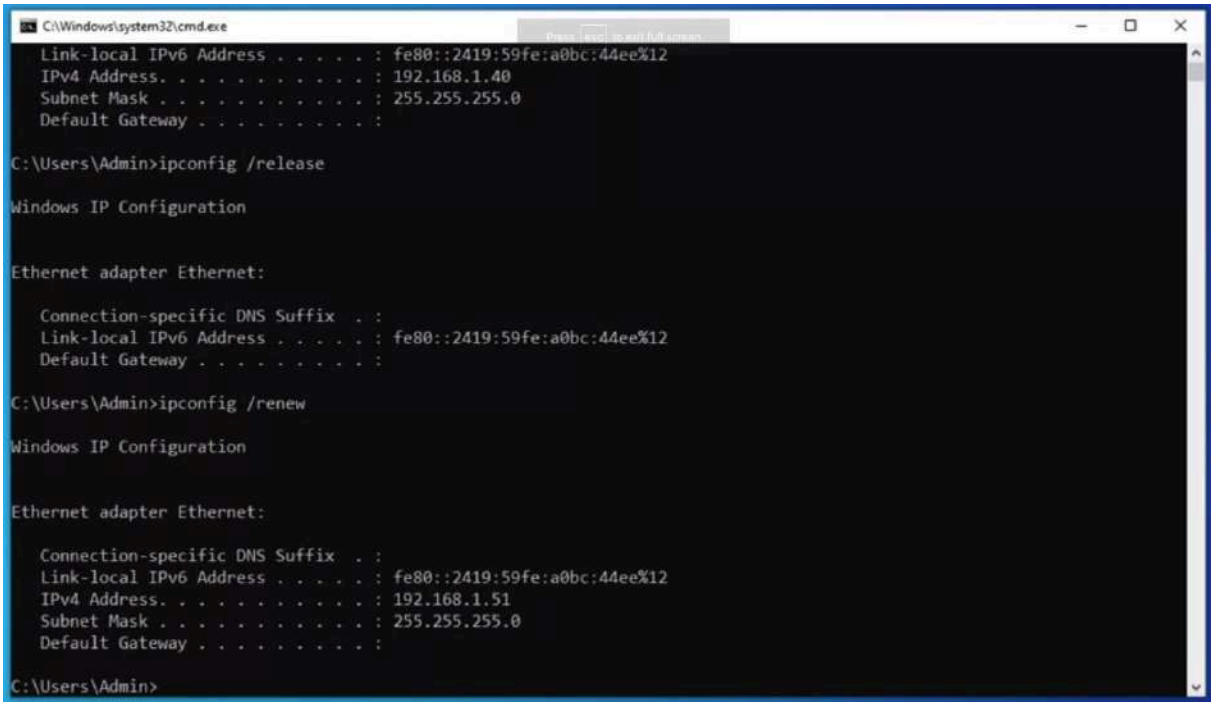




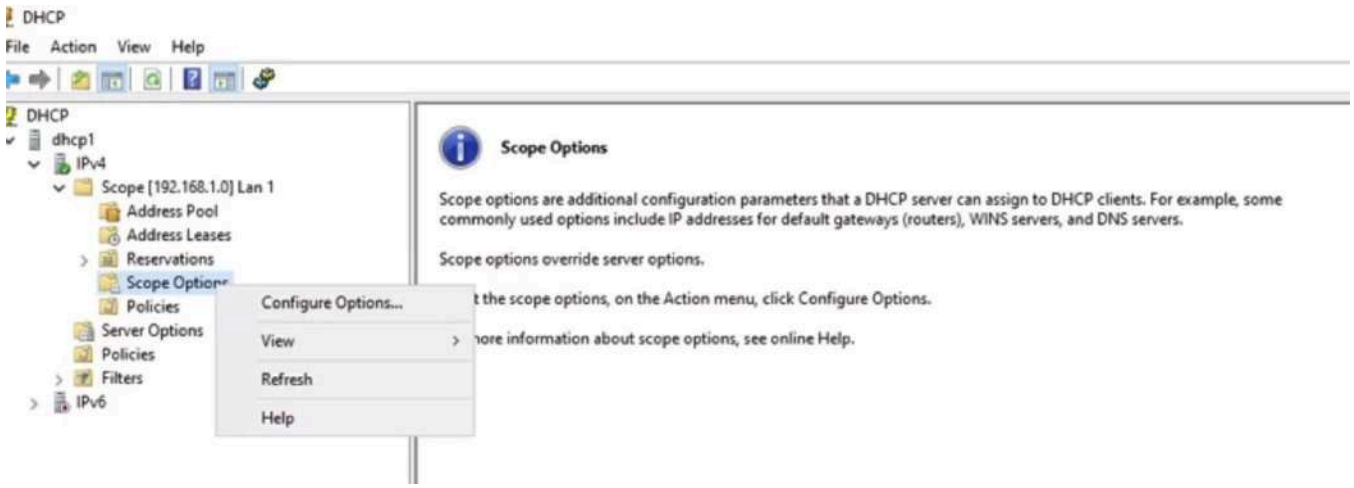
Then I'm just going to refresh. Now our scope runs 1-250, but the DHCP server will not give out addresses from 1-50 or 101-199. It can give out from 51 through 100 and 200 through 250. Exclusions identify static IP addresses in the environment. The DHCP server will not give out addresses that are on exclusion.

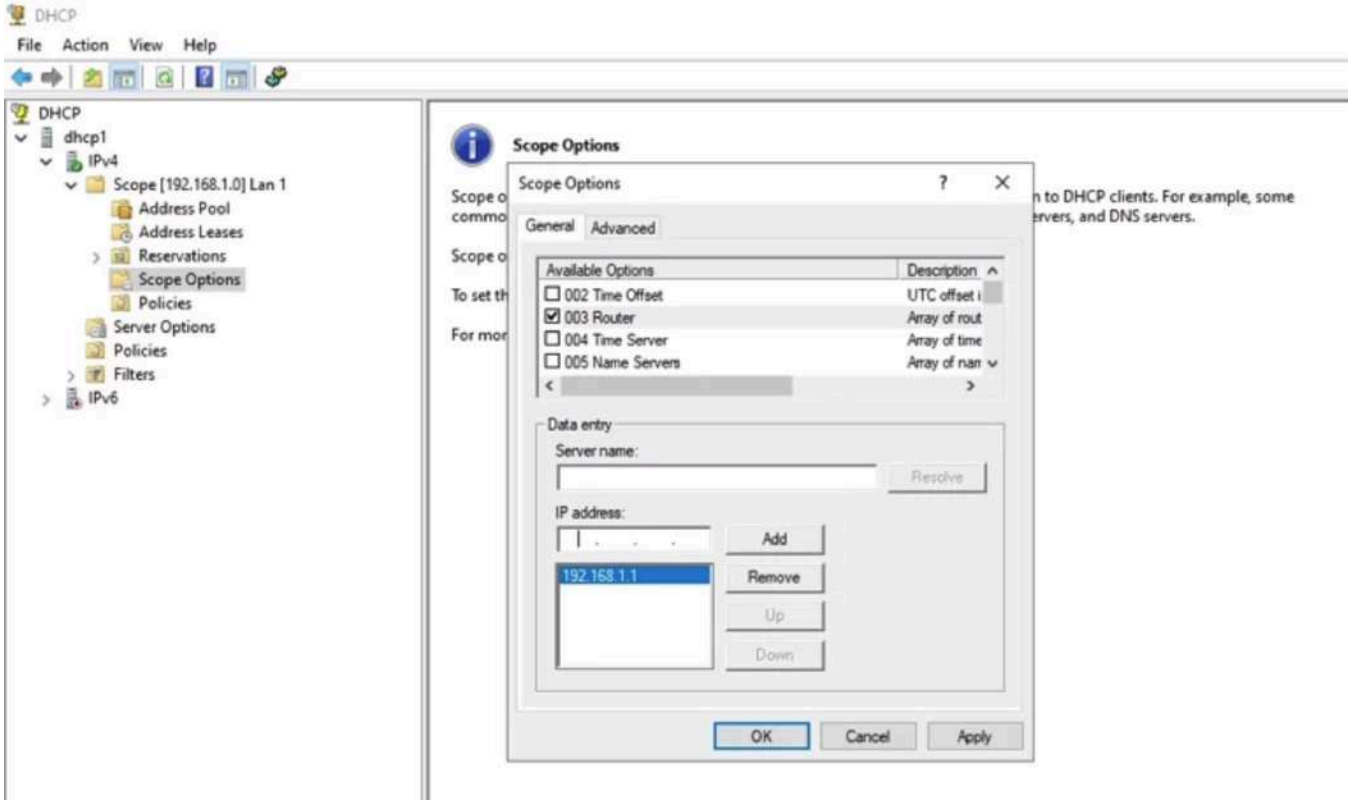


Let's go over to our client and take a look at what happens now. Back on our client, let's go ahead and release this address , then we'll renew. You can see it picks up 192.168.1.51, which is the first available address.

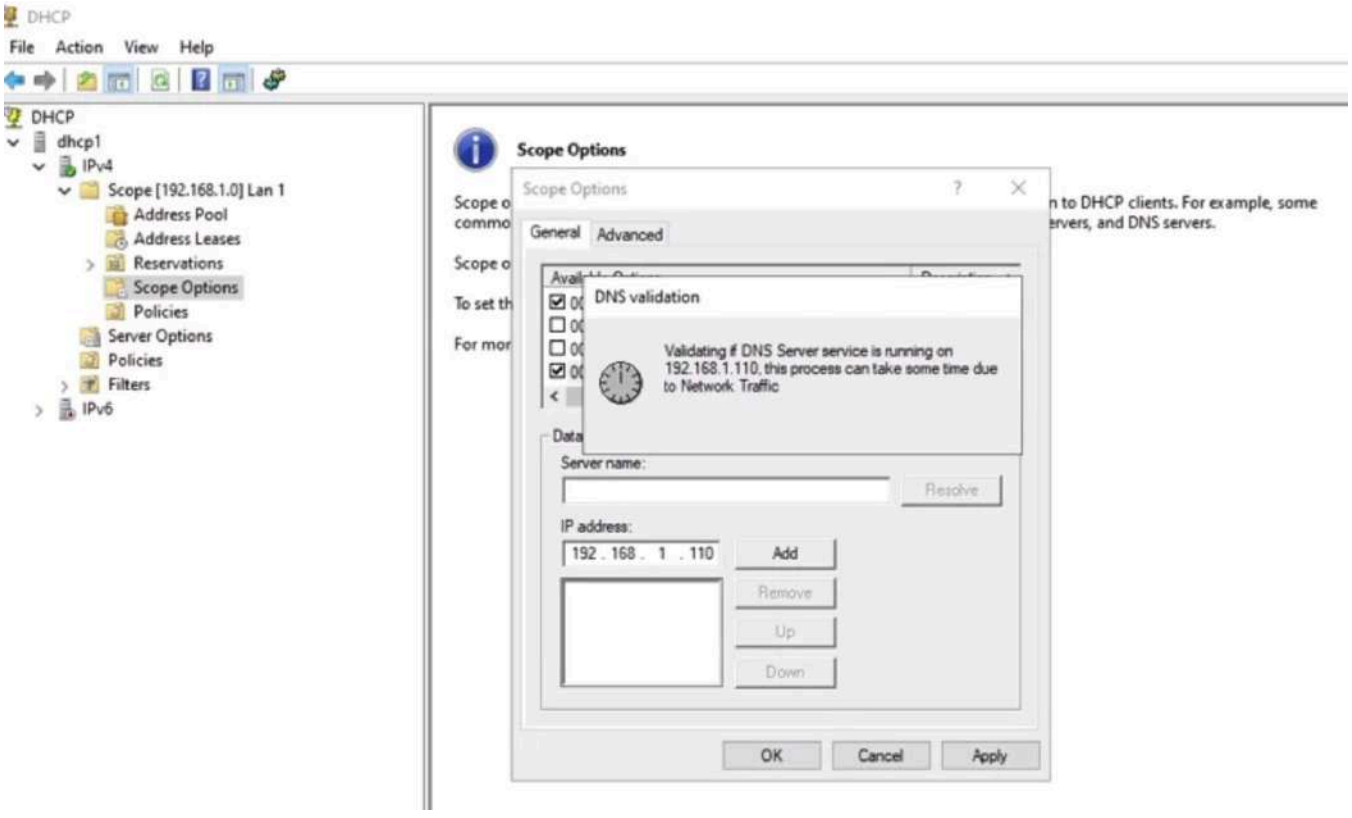


Now we're going to go back to our DHCP server. Exclusions are addresses the server won't give out. Another thing we see on the DHCP server are options. Options are extra information that should be given along with the IP address. I don't know if you noticed but a minute ago when we did ipconfig, that computer didn't have a default gateway. If it doesn't have a default gateway or a DNS server, it's not going to have Internet access. In order for it to have those things, we would need to add in options. So let's add some options. We're going to configure options 003, which is the router. Well, that's going to be 192.168.1.1.

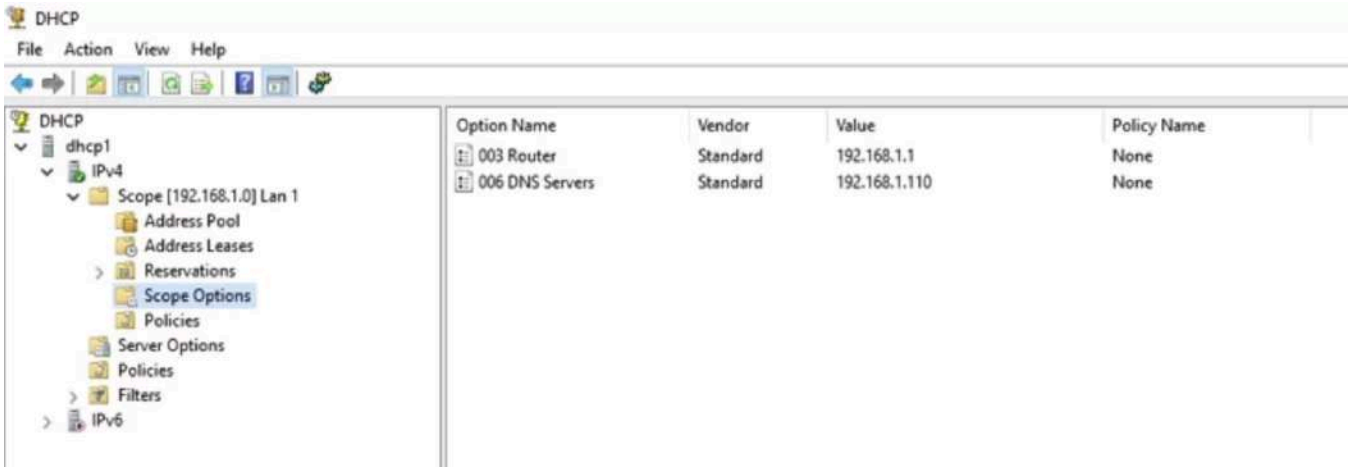




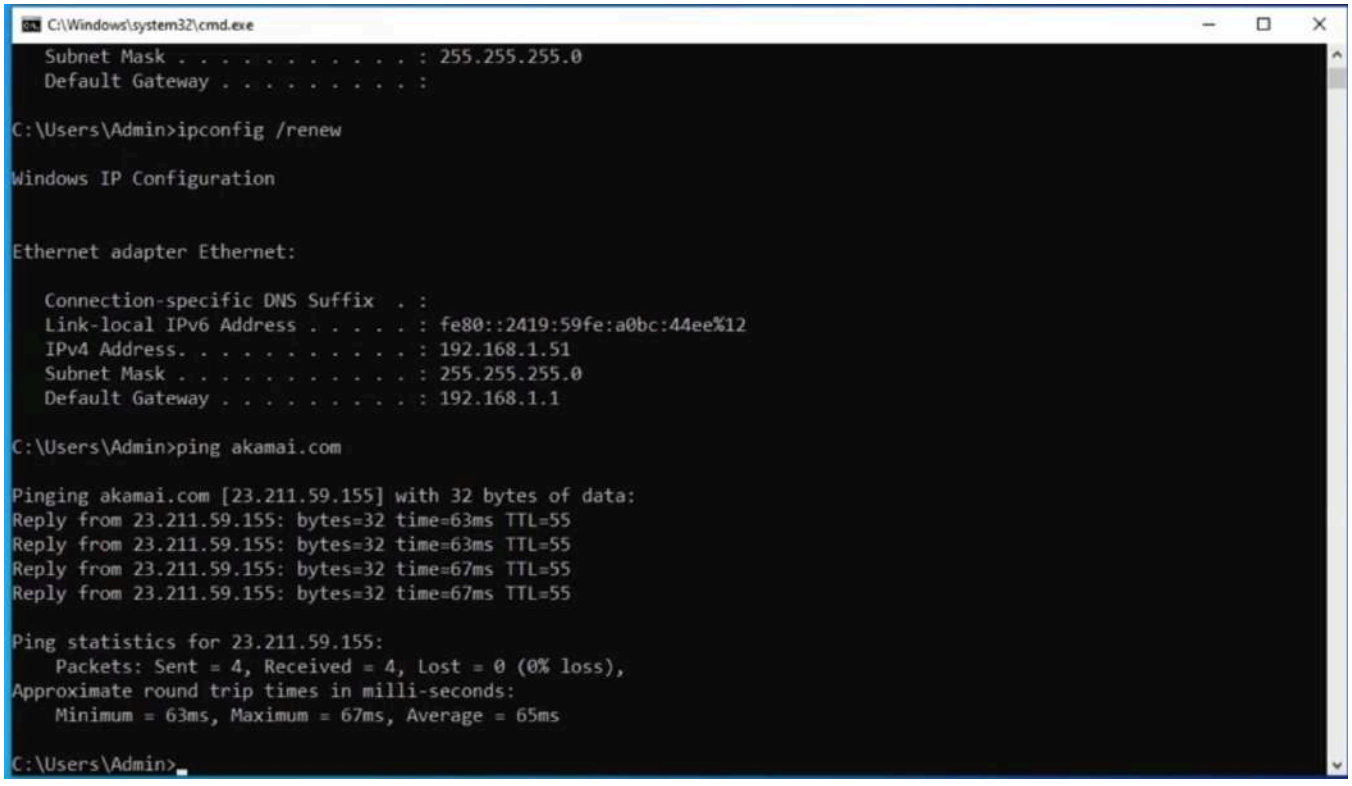
It's probably going to need a DNS server. The DNS server is actually at 192.168.1.110.



Perfect. Now, we have two options to give to the client. Let's go over to the client and renew to get those options and test Internet access.



Back on my client. Here you can see there's the results of the ipconfig /renew. There was no default gateway. Let's just go ahead and run our renew again. Now you can see it's got a default gateway. Let's just try a very simple ping, akamai.com. You can see that that's successful.



n this video we looked at some of the DHCP server settings. Specifically, we looked at scopes, which is the pool of IP addresses that the DHCP server can give out. We looked at exclusions which are addresses that the DHCP server should not give out and they're used to document static IP addresses in the environment. We looked at options, which is extra information that the DHCP server is going to give, along with the IP address and subnet mask, things like the router for the default gateway, the DNS server. Anything extra besides the IP address and subnet mask that the client needs would be done as an option.

DHCP Reservations

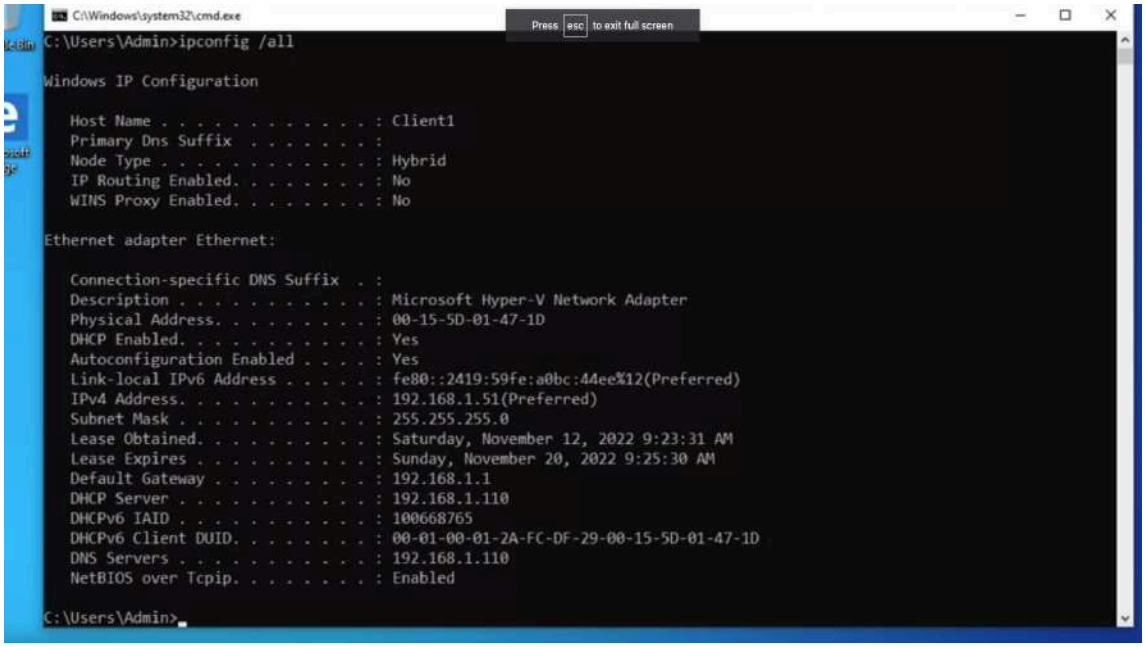
Sometimes you have clients that you want to keep set to obtain an IP address dynamically from DHCP. However it would be inconvenient if their address has changed. Right now, we're here on client one and it has obtained an IP address from DHCP. It's picked up 192.168.1.51. And that's probably fine for a client computer because if I move this to another network I wanted to change I want to leave it set. But a typical example of something where you don't want the IP address to change would be a printer or a copier. Now if you have something set to ATP, it always picks up the very next address that's available. So why would the IP address for a printer or copier change? Well maybe it had been picked up the 51 address and then it goes down for a while the printer is broke so that 51 goes back into the address pool and then something else get that address.

Then when that printer boots up you fixed it it's not going to pick up 51 because that's already been given to somebody else. The problem is with a printer you set up the clients to point to the printer at a specific address.

So if the printers address changes. Typically you would need to reconfigure all the clients. On the other hand, you might not want to set a static IP address because it's not critical infrastructure. So if you have to change the IP addressing scheme of the network it would be inconvenient to reconfigure the printer. You'd have to find the printer wherever it lives. You'd have to deal with the little LCD monitor. It's a pain in the neck, so that's where reservations are useful.

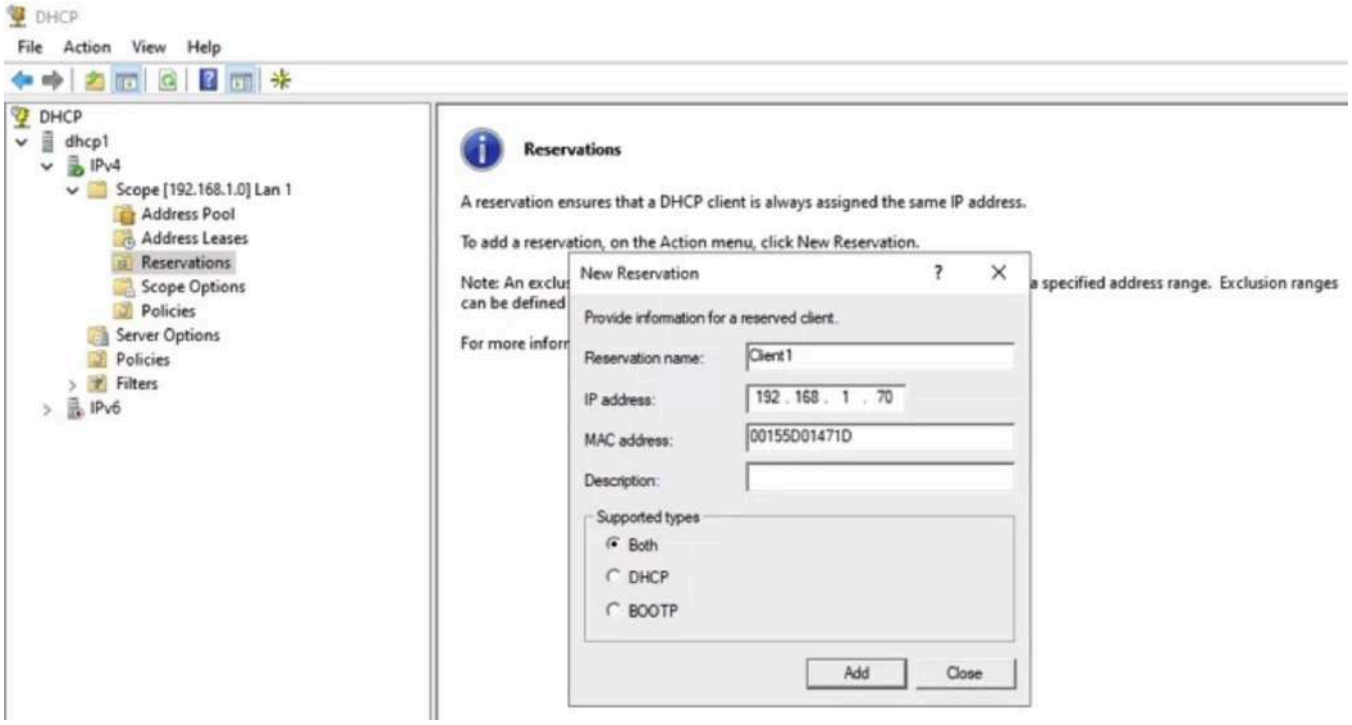
A reservation is a combination of an IP address from the scope in a Mac address. When you create a reservation, the DHCP server will only lease that IP address to the client with that Mac address. So we're going to be creating a reservation for client one.

What we need to take a note of is the Mac address of this client because remember the client uses broadcast that's the only way the DHCP server knows which client you're talking about. So that's this physical address here, the 00-15-5D-01-47-1D. That's the Mac address we're going to use when we create our reservation. So now that we know the clients Mac address. Let's go over to our DHCP server.



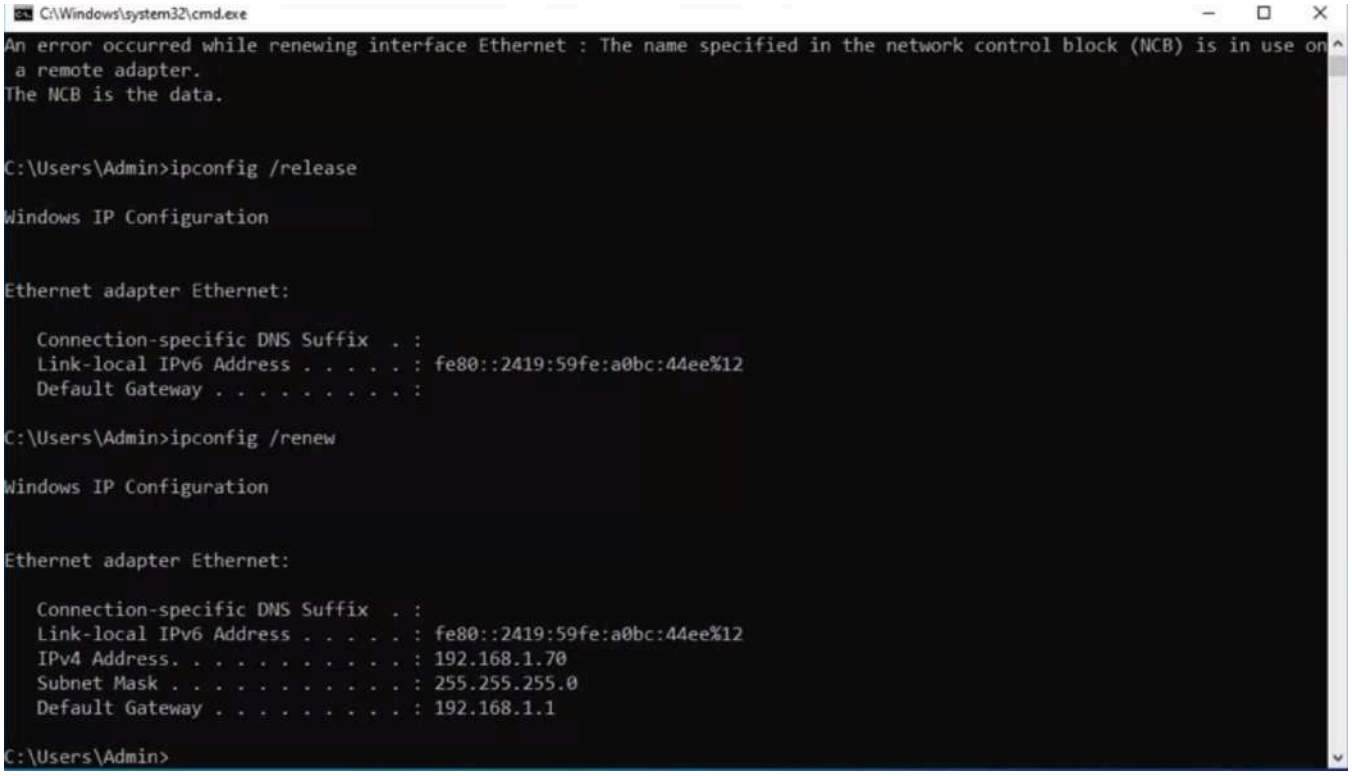
So we're here on the DHCP server and here's the scope that the client has obtained an IP address from. And if we go into leases you can see there's that 51 address to client one and you can see over here. It's got the unique idea of client one. Now I could just right click here and make this into a reservation but I want to do it manually because I want the client to pick up a different IP address. So it's easier to understand what's going on for the video. So we're going to go into reservations and say all right we're going to do a new reservation.

And this reservation is going to be for client one. But actually the DHCP server doesn't care what you name it. You could name it dog poop and it would be fine. We're going to tell it to reserve IP Address 0.70. And now we have to provide the Mac address of client one. Well that was 00-15-5D01471D. Have that written down. I didn't memorize it. Let's go ahead and add that and there's our reservation. So now the DHCP server is only going to give that 0.70 address to client one or whatever checks in with that Mac address.



Note that a reservation has to be a valid address from the pool. So if you have exclusions like we've excluded one through 50 and 101 through 199. Your reservation cannot come from the excluded addresses because excluded addresses are never given out. So a reservation is a good address from the pool that can be given out. But now it's been reserved for that client. It will only be given to that client.

So let's go back to the client and renew and see what happens. So back on our client we want to do our ipconfig/renew and it says hey you not too happy with that and it might be because I'm trying to renew but the DHCP service saying you can't renew 0.51 you're supposed to get 0.70. So I think the easiest thing to do is an ipconfig/release will let go of the 51. Now let's try our renew. It's much happier and there's the .70. I could have re recorded the video and taken out that error with the renew. But sometimes you see errors like that and it's a weird one. I wouldn't automatically know what that is if you're trying to renew. You get a weird error try release and then a renew. So you can see the client picked up the .70 which was the IP address that was reserved for that client.



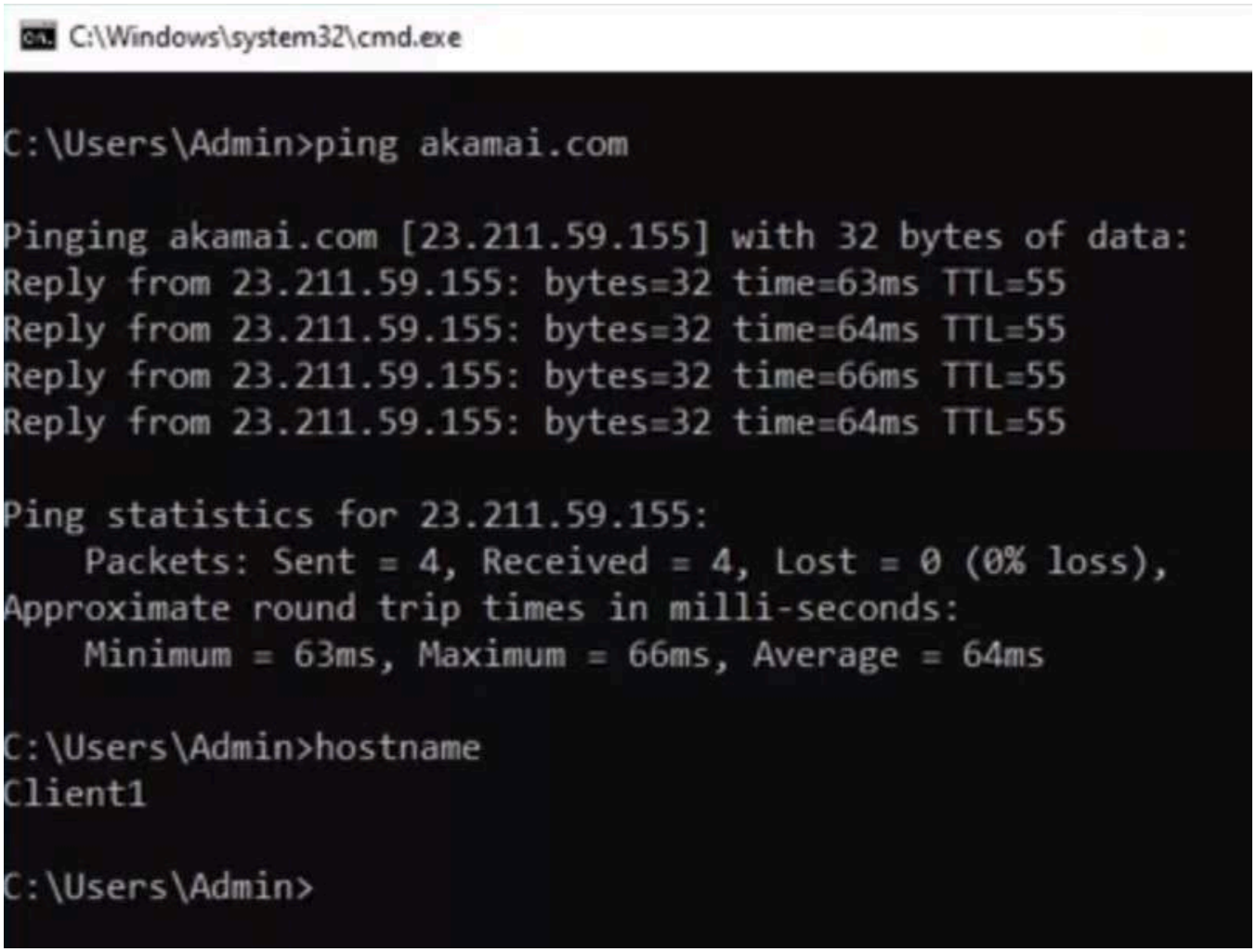
In this video we talked about DHCP reservations. DHCP reservation is an address from the pool and a Mac address of a client where the DHCP server will only give that address from the pool to that client. And therefore devices where you don't want to set a static IP address because they're not critical infrastructure. But you would prefer that the IP address not change.

Domain Name Services (DNS)

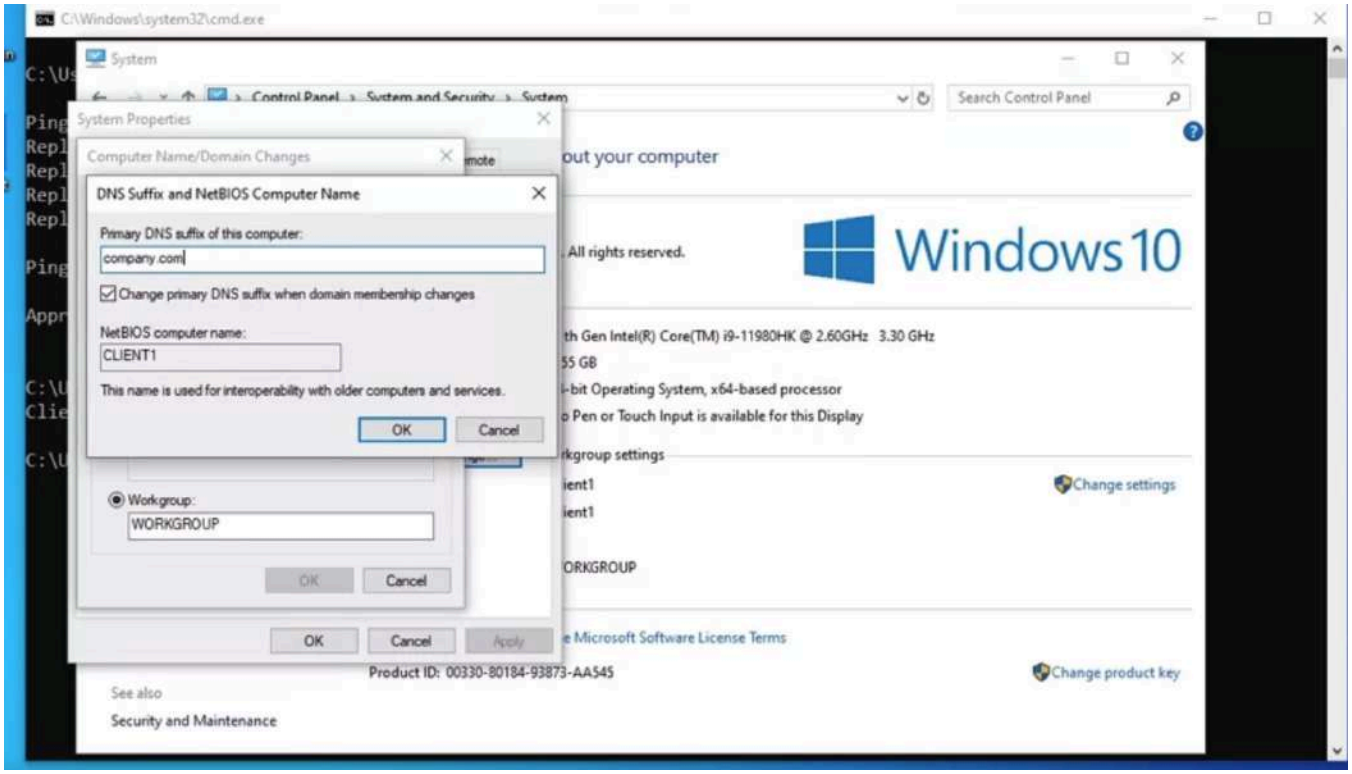
Host Names

Devices use IP addresses to communicate with each other. But IP addresses are not very friendly for humans, just like it's easier to remember a friend's name than their phone number, it's easier to type in a host name than an IP address. But we're here on our client, I'm just going to do a ping akamai.com. You can see that it actually makes the connection via IP address. Name resolution is when you resolve that name to an IP address. Now a host name is a unique name given to a node on a TCP IP network. In Windows we can check the host name using the command host name. The host name of this computer is client one.

When you combine the host name with the host domain name, the result is a fully qualified domain name or FQDN. Devices use name resolution to translate an FQDN to an IP address. It's name resolution that allows use names instead of IP addresses to communicate with other nodes in the Internet. Now, domain is a grouping of computers on the Internet based on the nature of their operations. Although there are several types of domains, some of the common ones are commercial, governmental, and educational. I'm sure you guys have seen dot com, dot gov, dot edu. Domain name is unique name that identifies an entity on the Internet. They might also be called site names, but domain names appear as part of a complete fully qualified domain name. When we're talking about fully qualified domain names, a period is used to separate each of the domain name labels. You can have no more than 63 characters per domain. They're not case-sensitive. The whole FQDN can be up to 255 characters in length.



Now this particular computer is not part of a domain, so it doesn't have an FQDN. If we go into system, we can see this, the computer name is client one. That's the host name. Then it says here, well, the full computer name is client one. There is no FQDN. Let's go ahead and give it one. I'm going to go into my settings, change the name, and I've got to go into more and say, well, this is going to be @company.com.



You've got to reboot if you want to change your computer name, okay.

Well, that reboot, we'll be right back.

My client has rebooted. Now if we go into control panel,

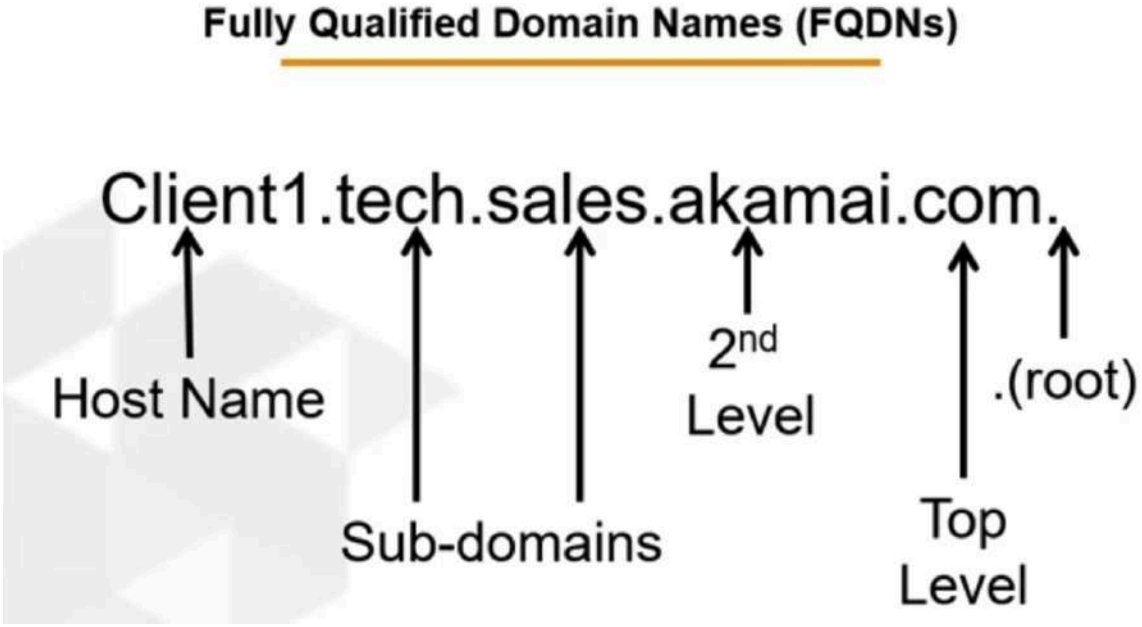
you can see that the full computer name is Client1.company.com, and that would be the fully qualified domain name of this particular computer.

A domain name like company.com or akamai.com, that identifies a collection of computers and devices on the network of a particular domain. A host name is a unique name that identifies a specified computer or device in a network. Host names are really subsets of domain names, but they're not the same thing. We'll talk more about that in the next video on fully qualified domain names. That's it for this video, we talked about host names and domains and got ourselves nicely positioned to talk about FQDN.

Fully Qualified Domain Names

So, FQDNs using standard.thelimited notation. So, that's a fancy way of saying that dot separates each section of the name, which you can see here, client1.tech.sales.akamai.com. The maximum length of the whole FQDN is 255 characters. Each dot delimited section can be up to 63 characters long, but you can see, if I had 63 characters in each of these sections, I would be over 255 probably. So we usually try to keep the names of each domain pretty short. And in FQDN, every time you cross a period, it's a different domain. Now, a network node can have more than one host name assigned to it, but its primary name is its host name. If it needs additional names, the other names or canonical names or C names, also known as aliases. But we'll explore that more in the DNS records videos. Although rarely type, you can see each F q d n has a dot on the right, which represents the root structure. So let's dig in a little bit deeper into this structure of a fully qualified domain name. When you type a fully qualified domain name, the name of the host is always on the far left. So originally when the internet started, it actually would be servers named, WWW, and such and such, a domain. On the far right is the period or the dot that represents the root servers on the Internet. And we're going to hear more about the root servers when we get to the name resolution lessons. Anything to the left of the root servers is the top level or first level domain name. In this example, it's the .com. Directly to the left of the top level of first level is the second level domain. Now, technically, the position to the left of that sales would be a third level domain. But any domains between the second level and the host, we just call

sub domains. And the reason is this, in most countries, all you can register is a second level domain. Any sub-domains under the second level domain belong to whoever owns the second level domain. For example, Akamai owns akamai.com, I can't go out and say, hey, I'd like to register shad.akamai.com. No, they can do whatever they want to the left of akamai.com, nobody else in the world can register a sub domain of that.

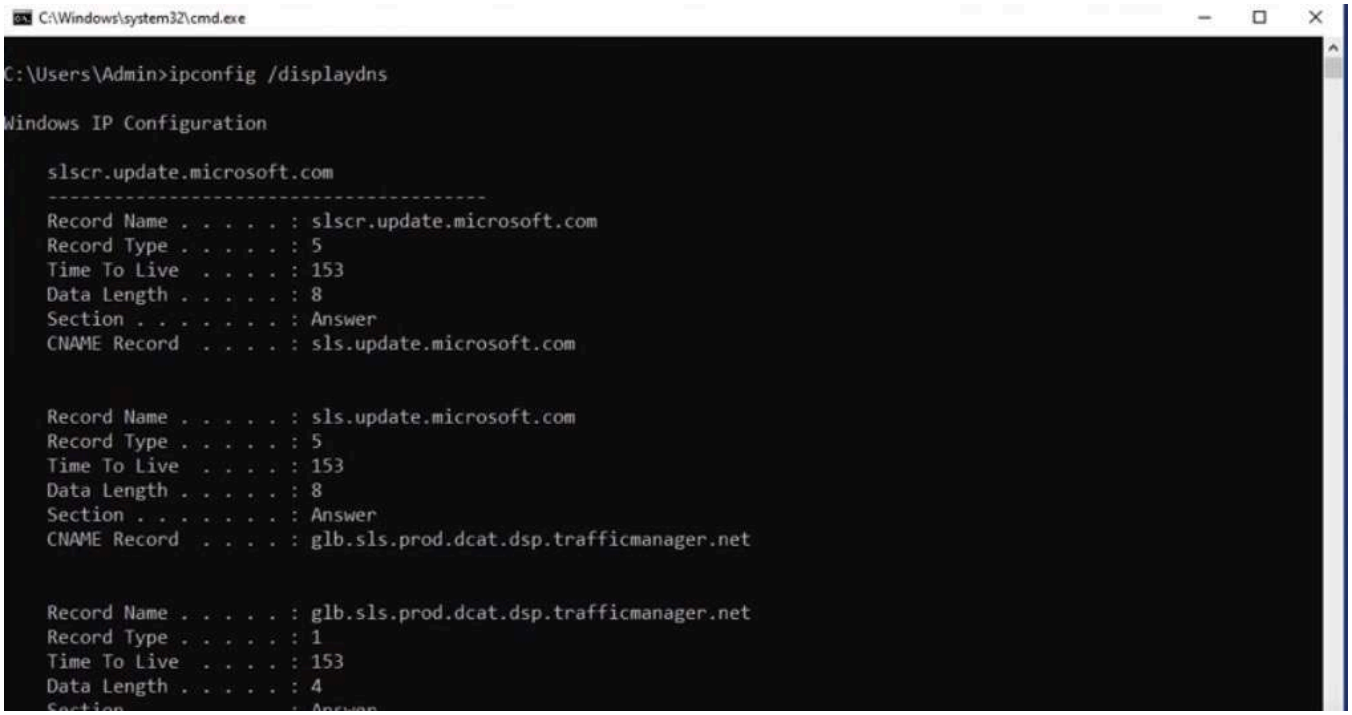


Now, there are a few exceptions. In the United Kingdom, for example, you don't register .com second level domains, you register a sub-domain of .co.uk. So a standard regular domain that you'd buy for your business would be a third level domain. But that's unusual in most countries, you're purchasing a second level domain. So that's it for this video. In this video we looked at the structure of fully qualified domain names FQDNs. We learned about the host name being on the left, the root of the DNS tree being on the right. And then we talked about how every time you cross a period, it's a different domain name, and looked at the different levels.

Hosts File

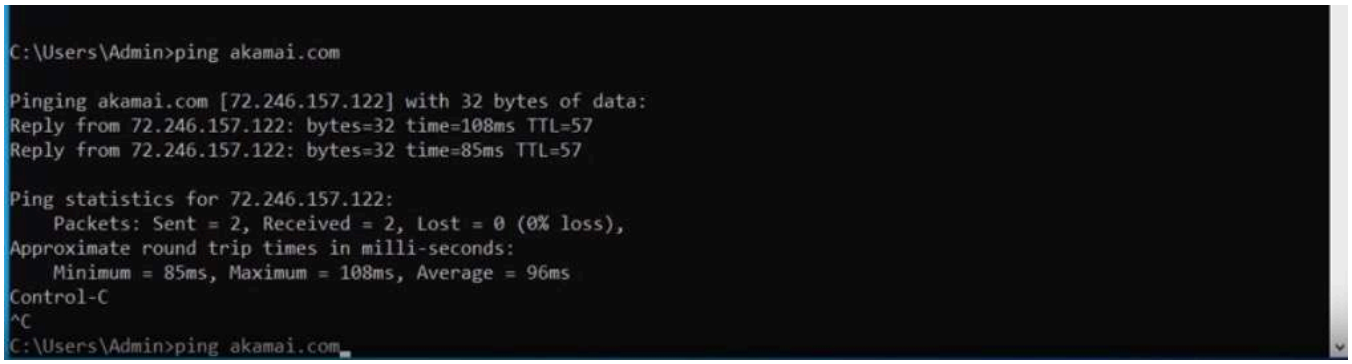
Up until about 1,000 nodes on the Internet, it used a central file called a host file to resolve names to IP address. The host file is literally a file named host and to this day, all the operating systems have a host file for backwards compatibility. Any entry in the host file is added directly to the DNS cache on the computer. The client always checks the DNS cache first before it talks to DNS. That means that entries in the host file will take precedence over information in the DNS database. This is because the client will always retrieve the information from the DNS cache first. If that information is in the cache, the client will not consult the DNS server. Let's take a look at this. Let's first start by taking a look at our DNS cache, and in a Windows machine that's ipconfig/displaydns, and you can see there's some stuff in here right now. Bunch of Microsoft Windows, very chatty. If I want to, I can clear out the cache, which would be IP config flush DNS. If I look at

the cache again, nothing in there.

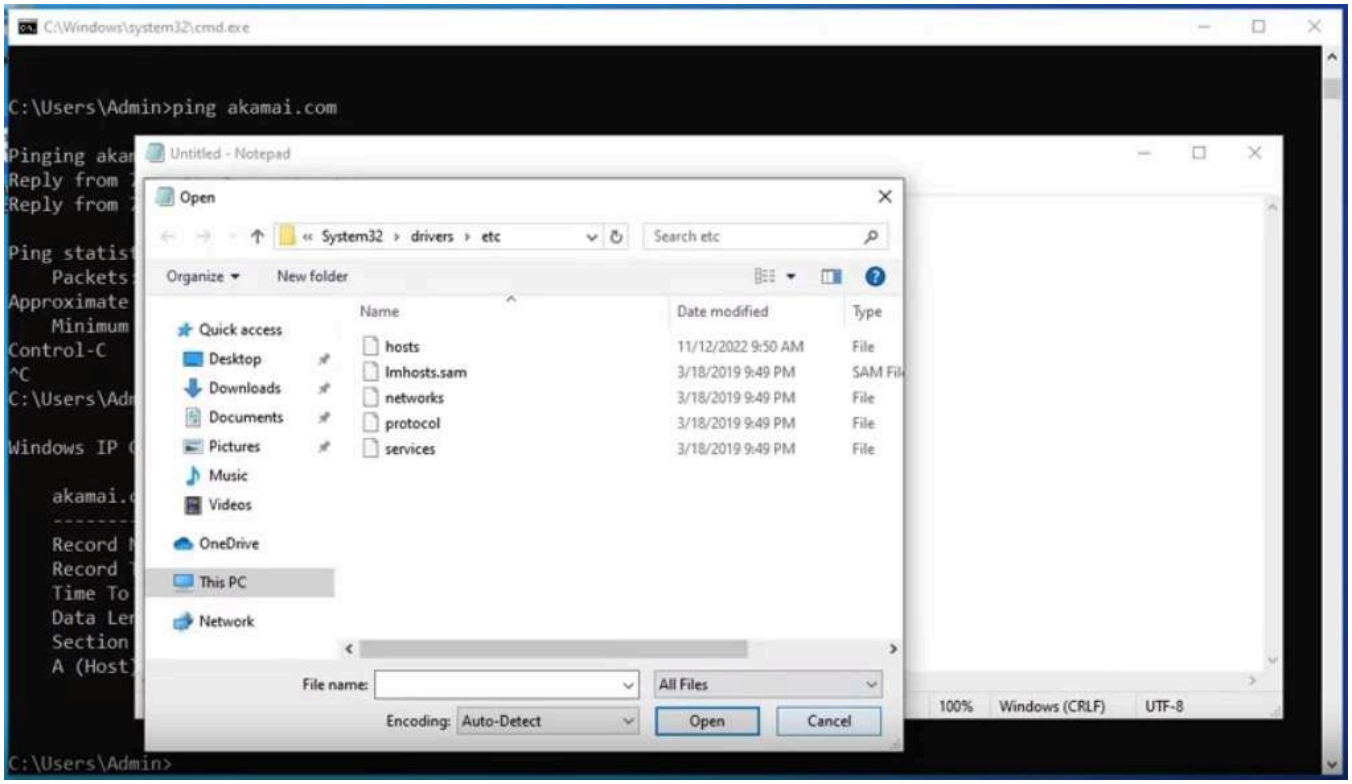


But I did something simple like ping akamai.com.

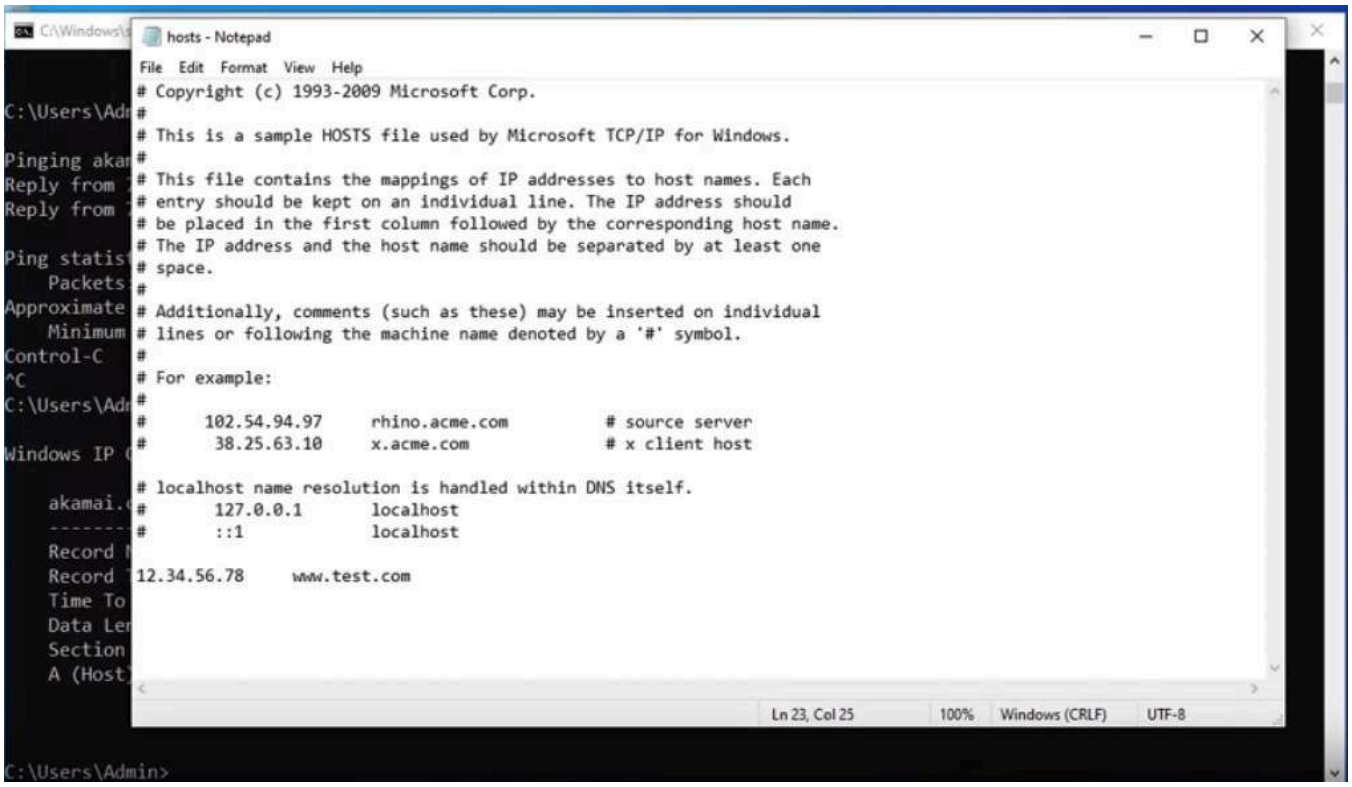
Now we display our cache. There's the Akamai record.



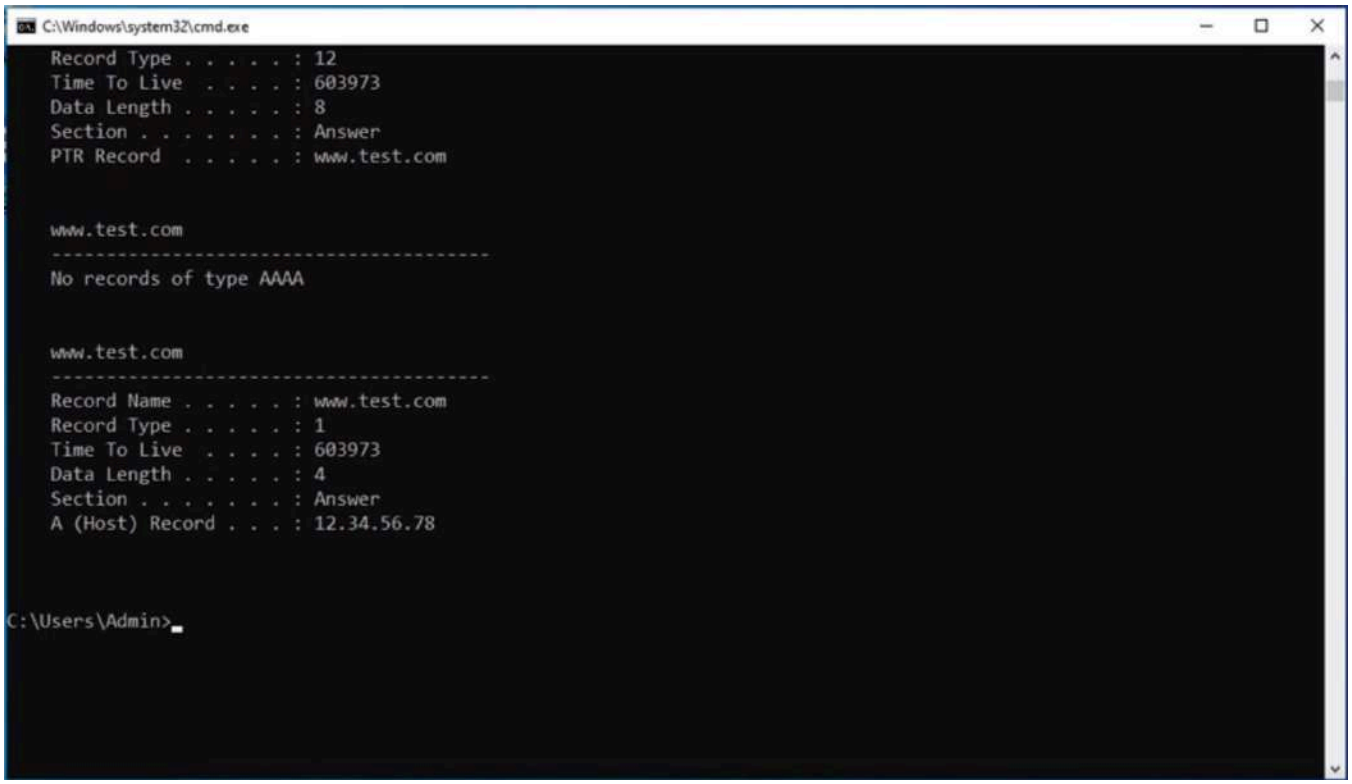
We're going to add an entry to the host file. Now, in Windows, this file can only be modified if you open it as an administrator and you have to edit in Notepad, and that's to protect it from different programs. Maybe malware adding a malicious entry to the host file. If you want to edit those file in Windows, you actually need to go in and open Notepad as an administrator and then do a file open. The host file is actually in Windows system 32 drivers, etc. You're saying, but Chad, there's no file in here. The actual name of the file is host. It doesn't have an extension. If you look at the dialogue box by default, Notepad says, well, I'm only going to open files that have a.TXT extension. The host file doesn't have an extension. If I change this over and say no, show me all the files, there's the host file.



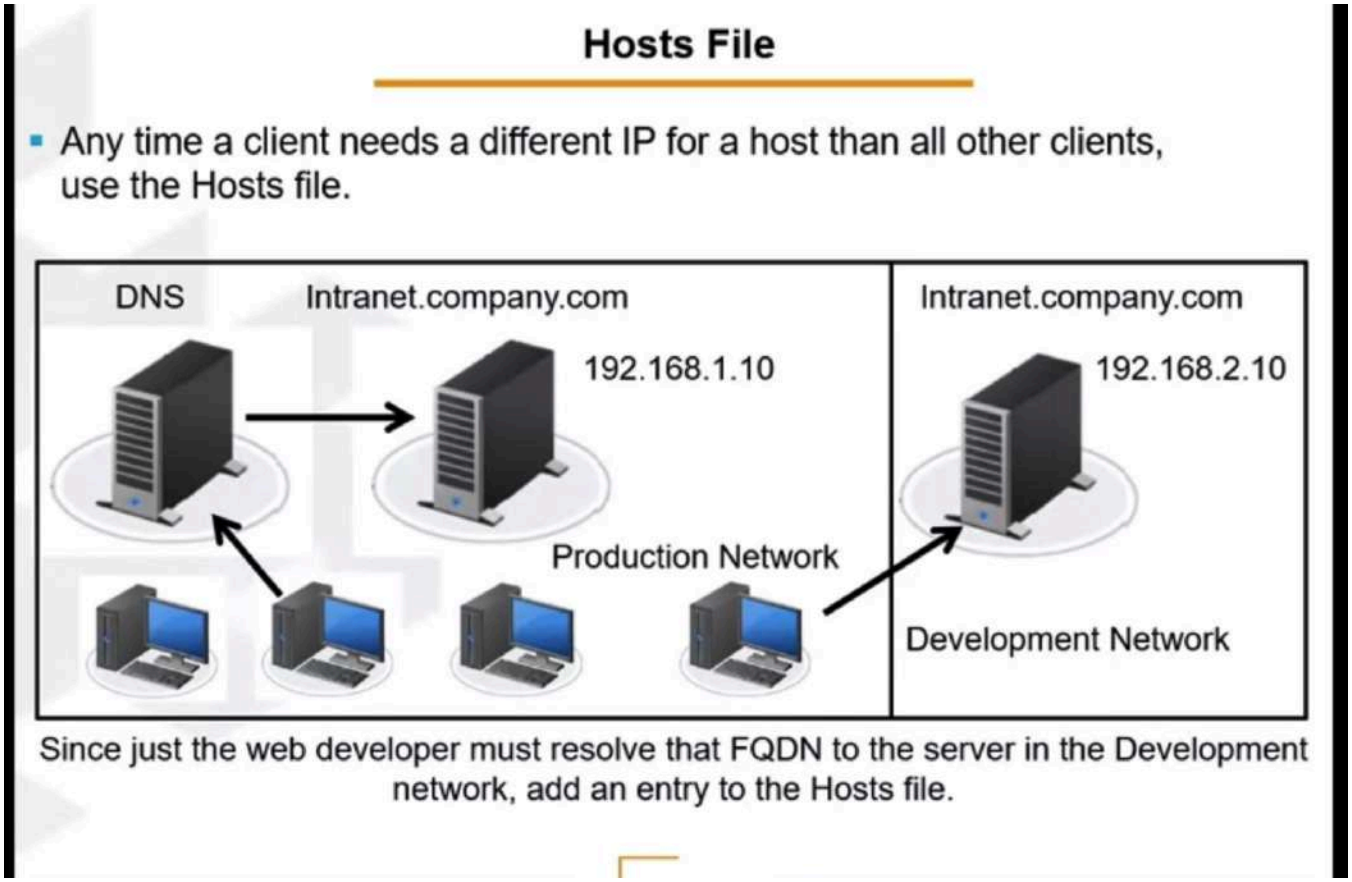
When we open it up, any of the lines that start with a number of pound sign, they're just comments to tell you how it works. Basically, what you would do is go down to a blank line, you type an IP address. Let's just make one up,12.34.56.78. Then you hit tab and then you type the host name you want to link it to. We'll just do www test.com and I'm going to save that. The minute I save it, that goes right into the DNS cache.



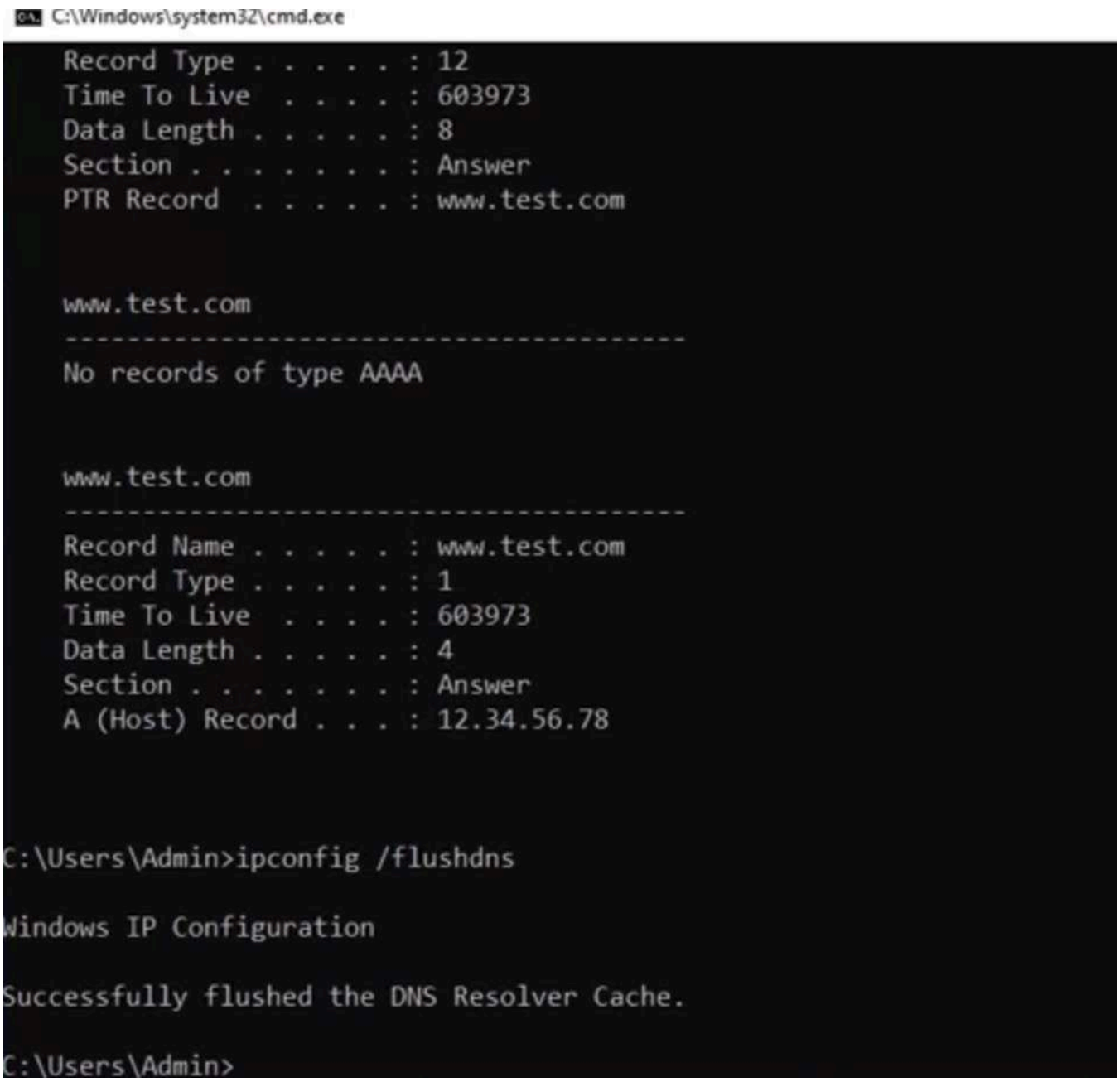
If I do my display DNS, then you can see there is that entry in there. Then that's all that's in the cache because the Akamai entry has expired. The host file can be useful if one or a small number of computers need a different answer than would be supplied by the DNS server.



That's the only legitimate reason why we should use this. I've seen technicians use it when DNS is broke. That is not a good reason to use the host files. Let's take a look at example. What would be a good reason to use it? A legitimate reason to use the host file is anytime a client needs a different IP address for a host than it would get from DNS. It needs a different IP address than all the other client. Here's an example. On the left, we have the production network, meaning the regular company network and we have a bunch of computers in here. When they talk to DNS and they ask for the IP address of Intranet.company.com, they need to get the IP address 192.168.1.10 because that's the IP address for the Intranet website. But there's just one computer which belongs to the web developer and when the web developer goes to Intranet.company.com, they actually need to get an IP address of 192.168.2.10 which is a development server over in the development network. They're going to upload updates the website, and they're going to go to Intranet.company.com and they need to go to the server that they're using to test these updates. The only way to have that one machine or a few machines go to a different IP address for that host, then what is contained in DNS would be to use the host file. But normally we don't expect anybody to put entries into the host file, so when you do this, you should document it.



Once had a student say to me, "Oh Chad, I really understand DNS. I was troubleshooting DNS for six months." I said, "Why didn't you call me and ask me. It's not you could've never helped me with this." He said what happened was a server changes IP address and none of the clients would go to the new IP address and I went through everything and DNS like I couldn't figure out why. But it turned out that the administrator before him had put an entry in the host file on all 60 computers to the old IP address of that server and so none of them were talking to DNS. When the server changes IP address, none of them went to the right IP address. Now, when you say you could've never helped me, hat's a challenge to me. Let me show you real quick how I could have helped them in two minutes to find that problem. Let's go back to our client. This www test.com is coming from the host file. Let's go in here and do an IP config flush DNS.



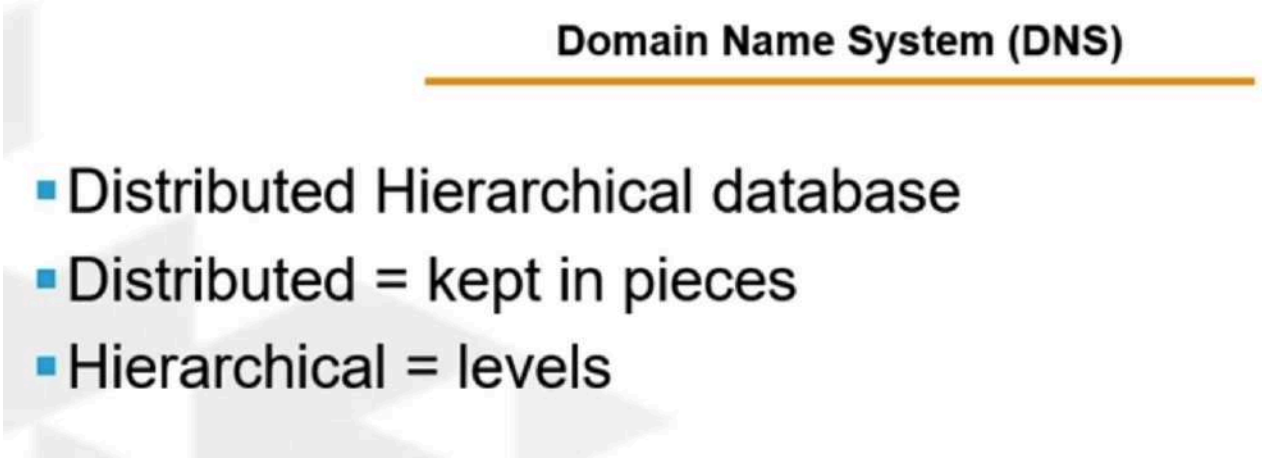
If you clear the DNS cache, but there's still something in it, the only place that could be coming from is the host file. Just a little tip for real life.

In this video, we looked at the host file, which is a file on every operating system for backwards compatibility. Any entry put into the host file goes immediately into the DNS cache, and it would override anything coming

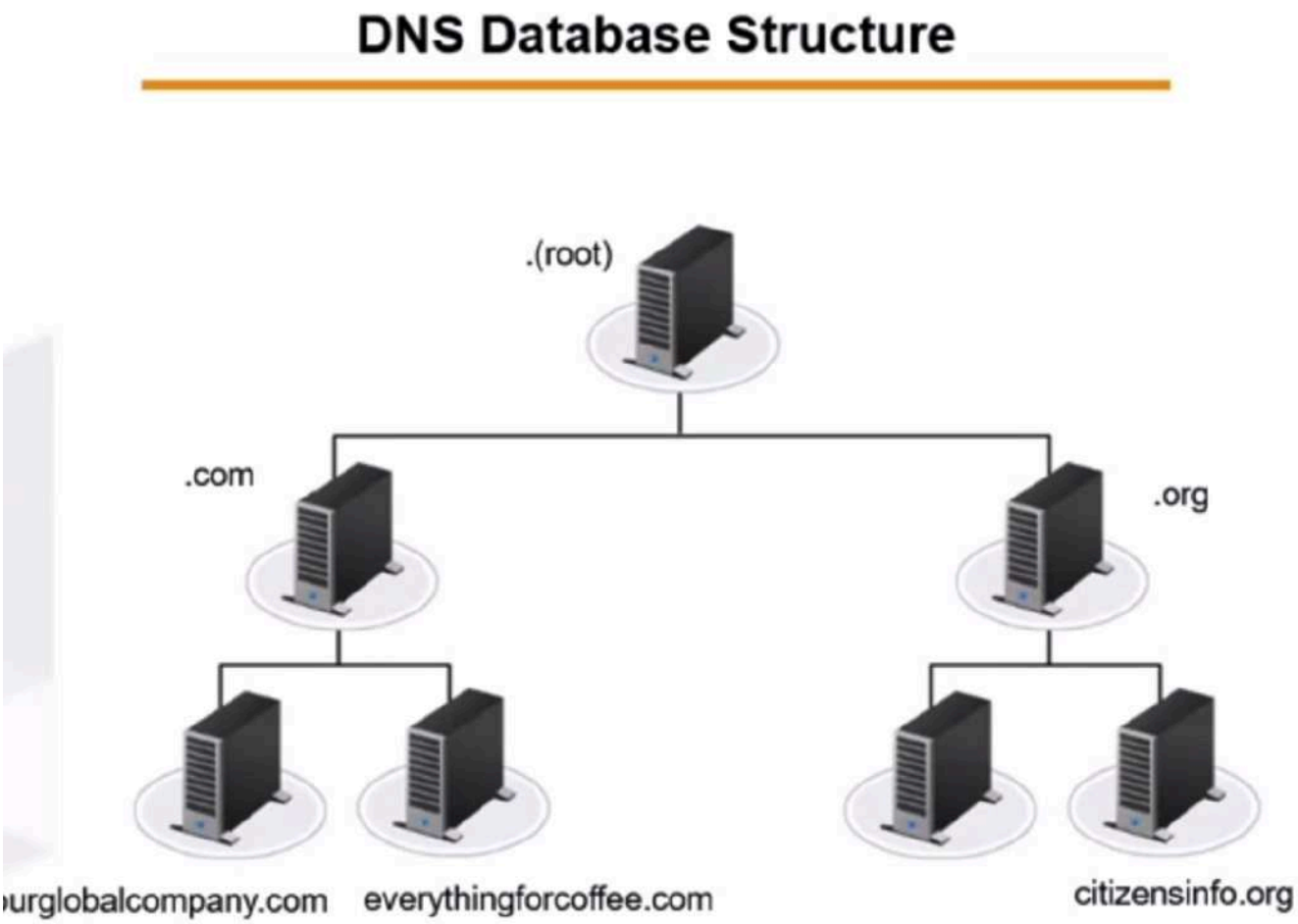
from DNS because the clients won't talk to DNS. We would use it just for that purpose. Anytime we need a client or a small number of clients to get a different answer than they would normally get from DNS, we can only do that using the host file.

DNS Overview

The two ways to do name resolution, the host file that we looked at in the previous video and the domain name system or DNS. So the internet actually used a centralized host file up to about a thousand computers. At that point, with the rate that computers were joining the internet, having just one file was not supportable. The domain name system DNS was created to provide a better solution. DNS is a distributed hierarchical database. And normally, I don't like words that are that big, but this is really the absolute best way to describe DNS. It's distributed, meaning it's kept in pieces. Each DNS server has just it's part of the whole DNS database, that makes DNS a distributed database. It's a hierarchical, meaning there are levels to the DNS database.



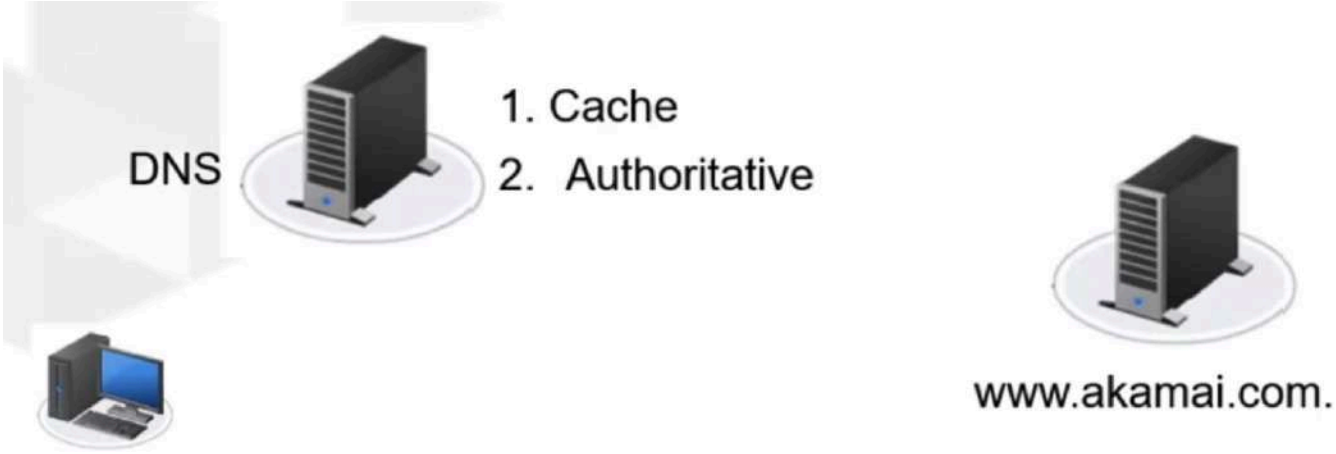
So when I think of DNS, I think of the phone book. When I was younger, used to get phone books delivered to your house, and in the United States we would have the yellow pages which would list all of the businesses for your particular area. So I live in Rhode island in the United States, and so we would get the phone book for that particular city or town. And that would have the list of all the phone numbers for the businesses in that area. Now you can deliver a yellow pages for the area to somebody's porch or their front doorstep. Imagine somebody trying to deliver a phone book for the world to your porch. It's not going to fit, right? Way too many numbers. So the telephone directory is also a distributed database, it's kept in pieces. Each little phone book has just its area. And it's the same with DNS. But with DNS there are also hierarchies. So, this is a little bit about the DNS database structure. We have the route at the top, and then we have the first level or top level domains.com, .org. And then below them, we have the second level of domains. And we'll see a little bit more of this in the next video about name resolution. Each registered domain owner needs to use a DNS server to store the DNS records for that domain. And when the servers resolve names to IP addresses, they navigate the hierarchy to find the right DNS server in the DNS tree. That's what they call this whole thing, the DNS tree that has the record that's needed. So that's just a high level DNS overview, looking at DNS as a distributed hierarchical database. In the next video, we'll really dig into name resolution.



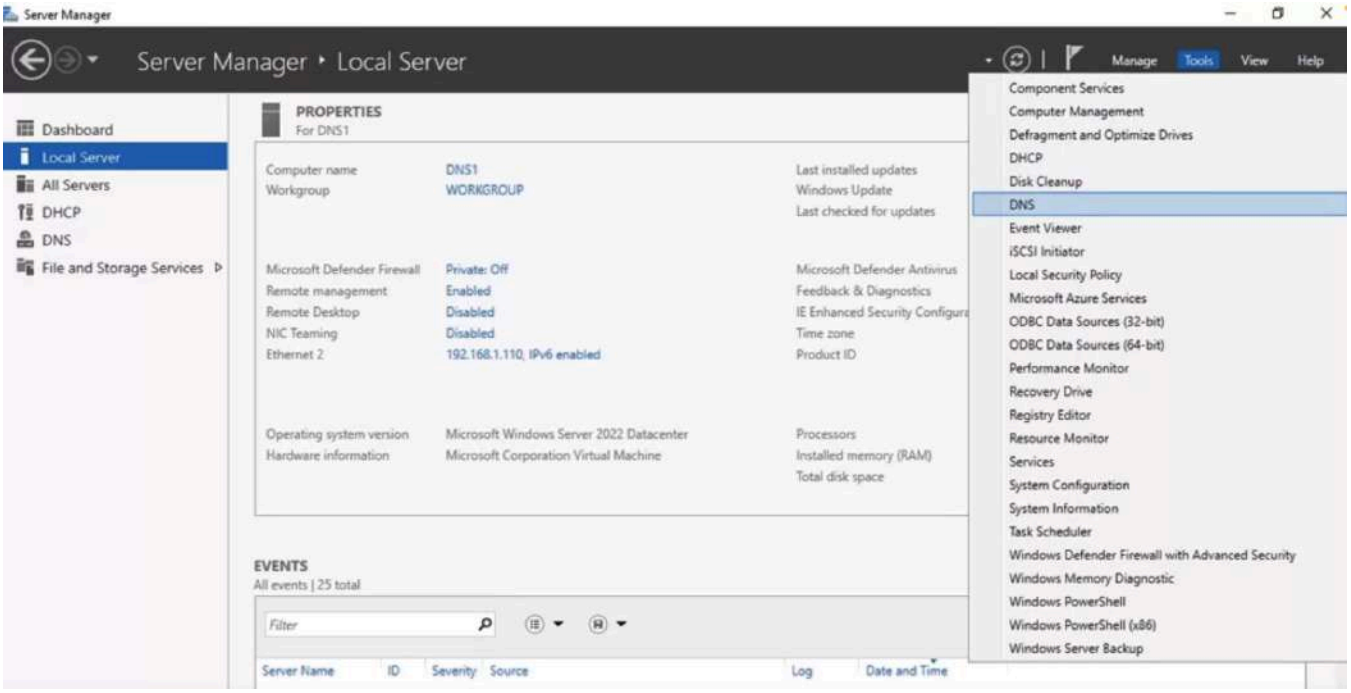
DNS Name Resolution Part 1

Let's take a look at what actually happens when the client wants to resolve a name. We know the first thing that the client is going to do is check its cache. What's going to be in the cache are names that it's gotten from DNS before and anything coming from the host file. If the IP address it needs for that host name is not in the cache, the client will then contact the DNS server.

The first thing that DNS server does is look in its cache. Maybe it's already resolved that name before and then it can just answer from the cache. If not, then it wants to know, am I authoritative for the domain? Am I the DNS server that has the records for this particular domain? If so, I can just essentially ask myself.

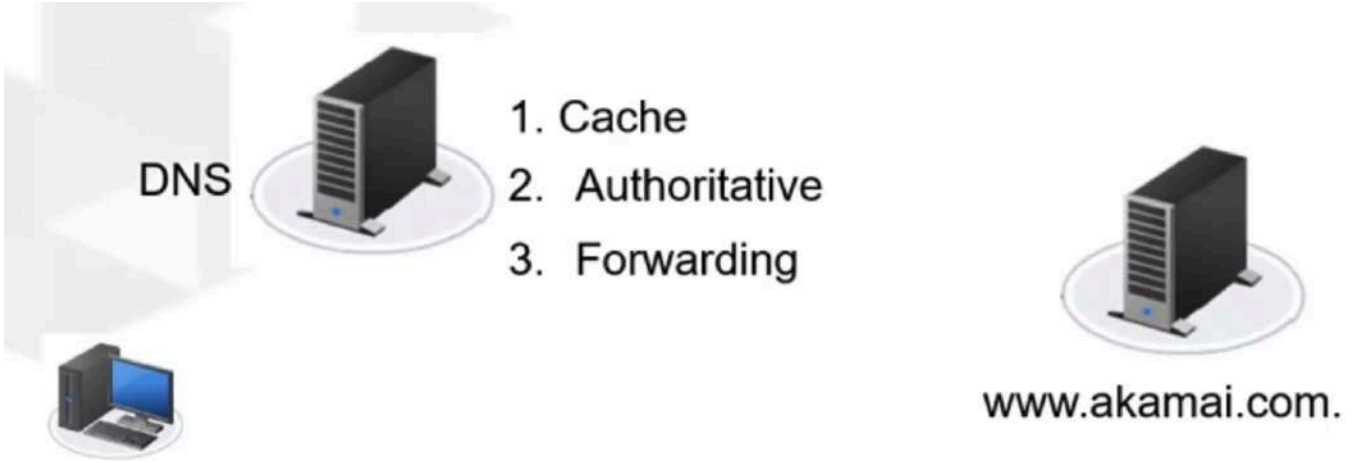


Let's take a look at the DNS server. This is my DNS server, DNS1. We're going to go into the DNS management console. This is a Microsoft DNS server, but they all work very much the same.



You can see in here that it has a zone, company.com. This DNS server is authoritative for company.com, meaning it hosts the records for that particular domain. If a request came in to this DNS server for something.company.com, it could just answer from its own database. If the server isn't authoritative for that domain, so in our example, this server is not the DNS server for akamai.com, then it looks to see if forwarding is set up.

When DNS servers are set up with forwarding, it means they're going to forward the request for any record for which they're not authoritative to another DNS server that's going to do the work of resolving that to an IP address.

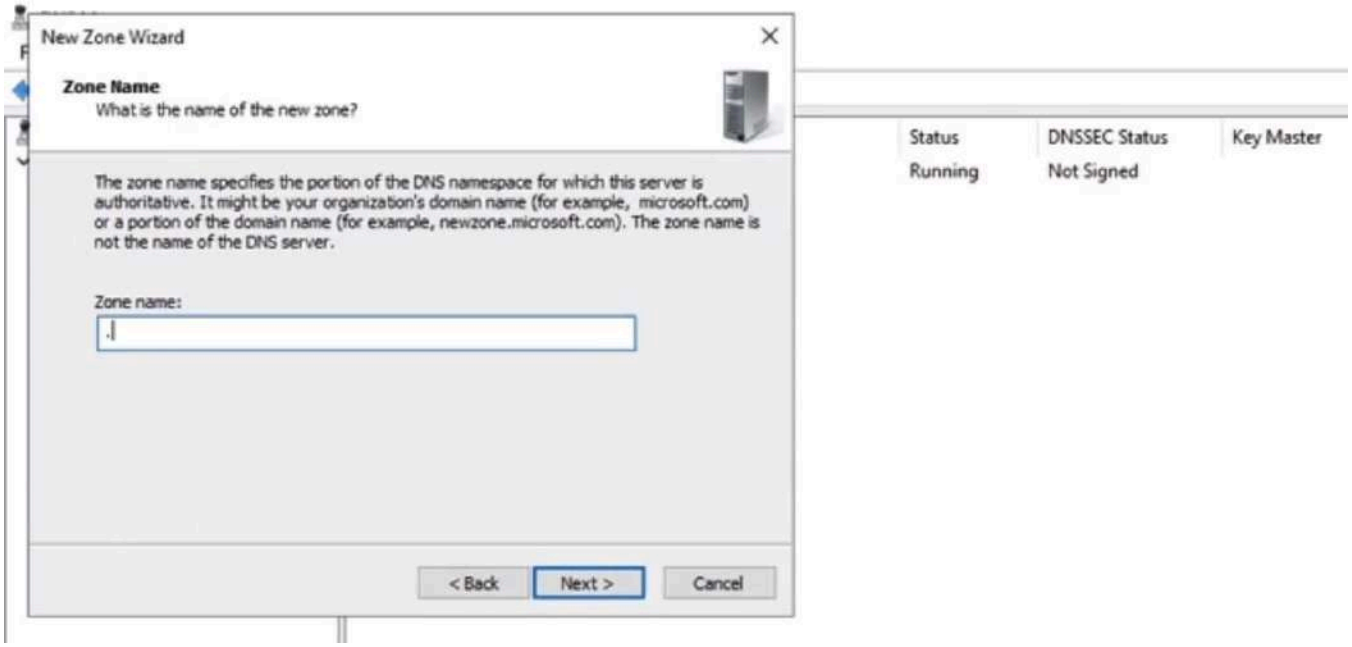


Let's take a look at an example. Suppose we have a setup like this. We have an internal network with four internal DNS servers here inside the network. Those DNS servers are going to provide name resolution for the internal client. If all of those name servers are going to resolve names on the Internet, then all of them would have to talk to any DNS server out on the Internet. But maybe for security purposes, we're not very comfortable with that. Instead, what we do is out here in the demilitarized zone or DMZ. If you're not familiar with the DMZ, we're going to talk about that in the security module. It's just a little network out here that's semi-private. We'll put a DNS server out here. When the servers on the internal network get a request for name for which they're not authoritative, they can forward that to the DNS server in the DMZ. That way we only open up a very small hole in the internal firewall between the internal network and the DMZ. The DNS server in the DMZ can talk to any other DNS server on the Internet. Again, this is not as wide of a hole in the external firewall because it's just one DNS server talking to any DNS server on the Internet rather than four. Here, we're getting a little bit better security because we're regulating which servers are allowed to contact the Internet. It also could speed things up a little bit.

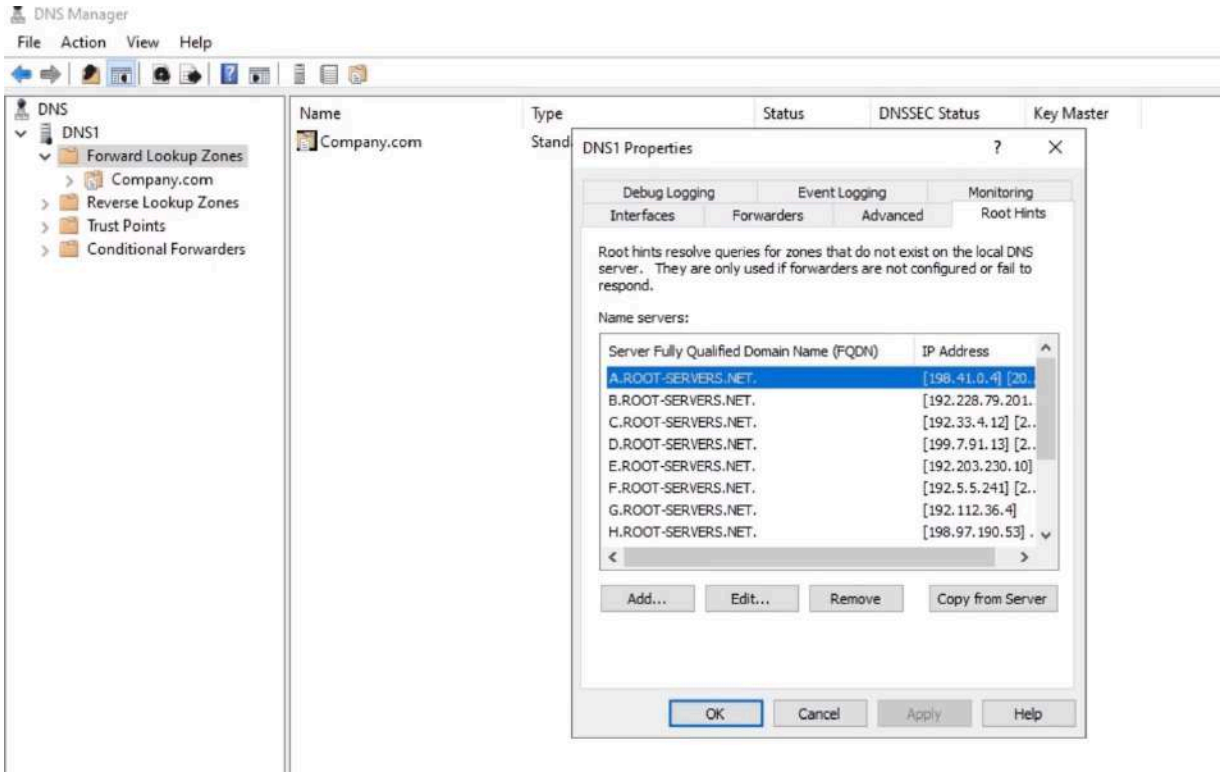
Because let's say this top server needs to resolve a name on the Internet like google.com. It's going to forward it to the DNS server in the DMZ. That DNS server is going to figure out what the IP address is. Now let's say some other DNS server needs to resolve google.com, like let's say the bottom one here. Then when it asks the server in the DMZ, that DNS server can just answer from its cache. Forwarding can either be done for security or to speed things up.

If forwarding is not set up, then the server will go to root hints. Root hints have the names and IP addresses of the root servers on the Internet. Let's take a look at root hints.

In the DNS overview video, we looked at the root servers as being represented by a period or a dot that you type on the far right of the fully qualified domain name. The root servers have a zone for the root of the DNS tree and it's literally a zone. That's what these things are. Like company.com, this is a zone, and it represents the records for which is authoritative. A root server literally has a root zone and it's just a zone named with a dot. Here we can actually create one right now. Let's name it with a dot.



Congratulations on a root server on the Internet. In this root zone would be information about those top-level domains and we'll get to that in the next video. I'm going to go ahead and delete my root zone because if I'm a root server, then this DNS server is not going to connect to the internet. I'm the top of the Internet for the DNS tree, I don't need to talk to anybody. Let's go ahead and delete this root zone. Are you sure? Yeah. It says, hey, you deleted the root zone. Do you want root hints back? Yeah. This is root hints. The minute I install a DNS server, whether it's in Microsoft or any other vendor, it's going to come loaded up with the names and IP addresses of all the root DNS servers on the Internet.



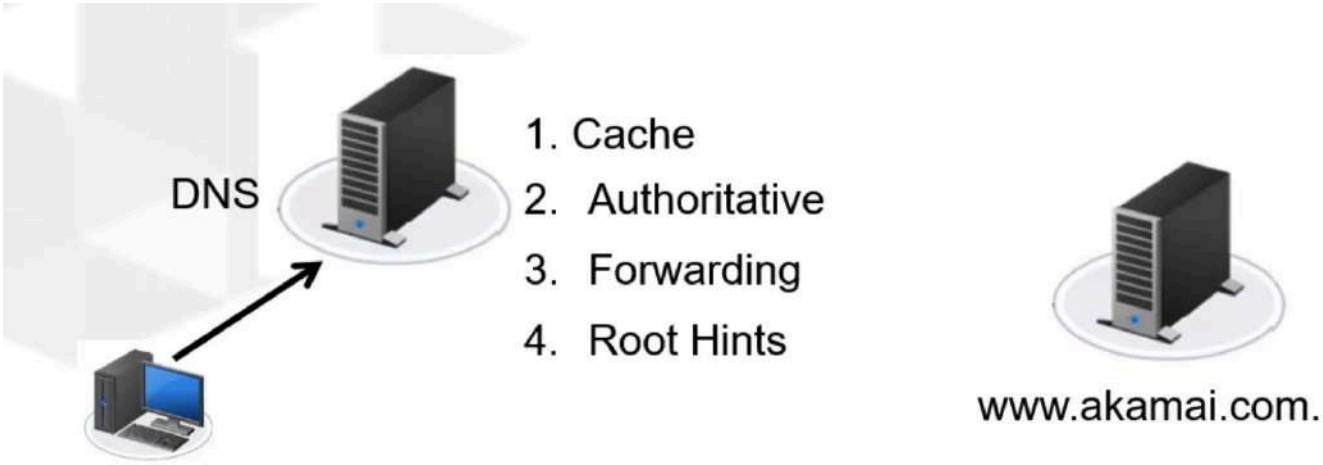
Just make a little note here. This is the A server and you can see it's got this 198.41.0.4 address. Then you can go out and look at this on your own if you want. You would go to root-servers.org.

But if you scroll down below the map, it's got the names of all the servers. The A server is actually run by various sign. These are all the places where it has a copy of that server and there you can see that same address that we saw in root hints, the 198.41.0.4 address. The root servers don't change their addresses very often. It's been that address as long as I've been coming out here, which is more than a decade, might even be as many as two decades to be honest. As soon as you install DNS, it's got the names and addresses of the root servers so that it can find the DNS tree out on the Internet. Once the DNS server gets to root hints, it's now going to begin following that fully qualified domain name, starting with the period at the far right-hand side that represents the root and moving to the left until it finds the DNS server that is authoritative for the domain that the record is in

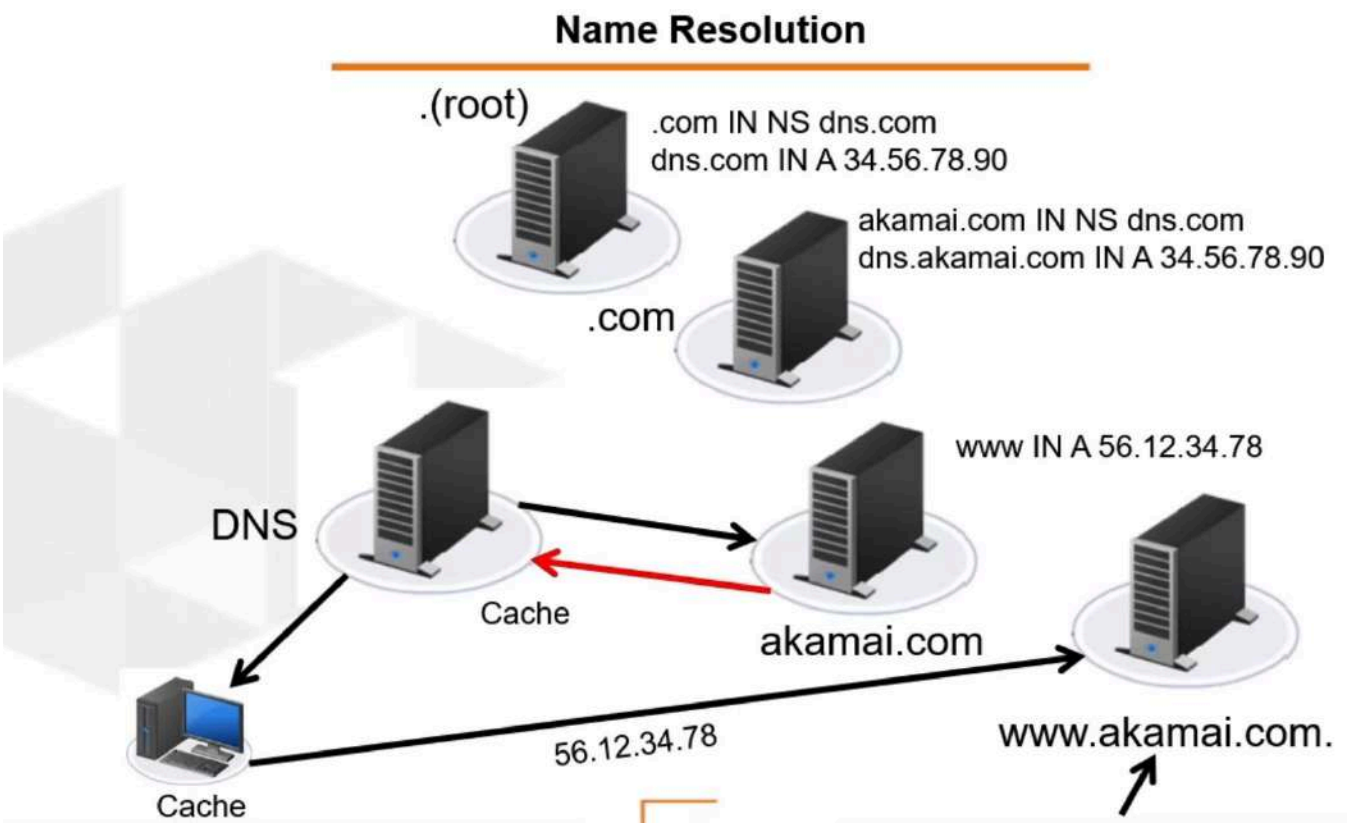
We looked at the beginning of name resolution. We looked at the client checking its cash, sending out to the DNS server. We went through the four different places that the DNS server will check to see if it can resolve it, ending up with root hints where it's going to begin contacting the root servers on the Internet to resolve that name. In the next video, we'll pick up and see what happens with the root servers and how the DNS server finishes resolving that name on the Internet.

DNS Name Resolution Part 2

In part one, we looked at the client checking its cache, forwarding the request to DNS server. We saw that the DNS server will check its cache, then see if it's authoritative for that domain. If forwarding is set up, it will forward the request to another server, and that can be done either for security or to speed up name resolution. But finally, if none of those things are in play, it will consult root hints which gives it the names and IP addresses of the root DNS servers on the Internet, and now it's going to begin following that fully qualified domain name from the period at the far right that we normally don't type, but it's always there that represents the root of the DNS tree to find the authoritative DNS server in the DNS tree.



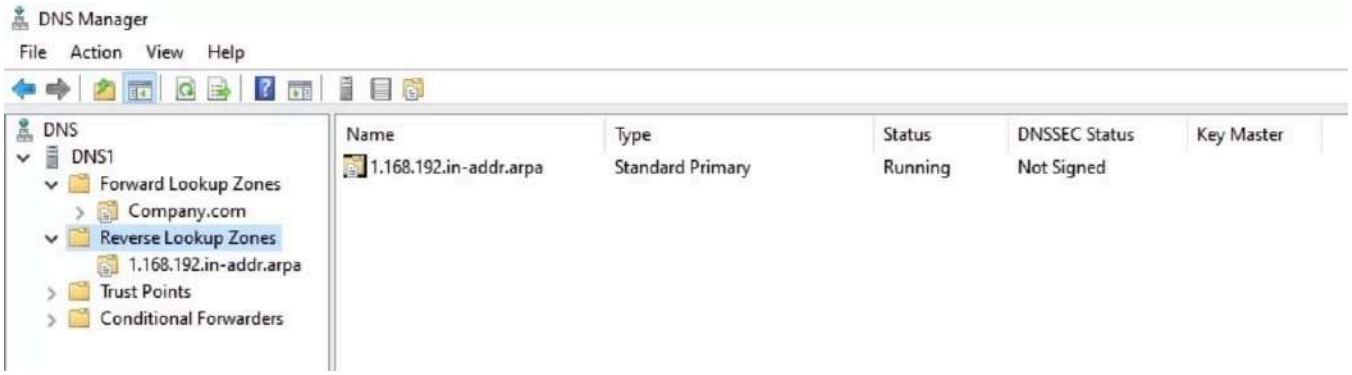
My DNS server trying to resolve www.Akamai.com dot, is going to contact the root servers. You can see down here in the bottom right-hand corner, we're at that period that represents the root on the far right of the fully qualified domain name. Now the root has information about the top-level domains, and this is technically called a delegation,



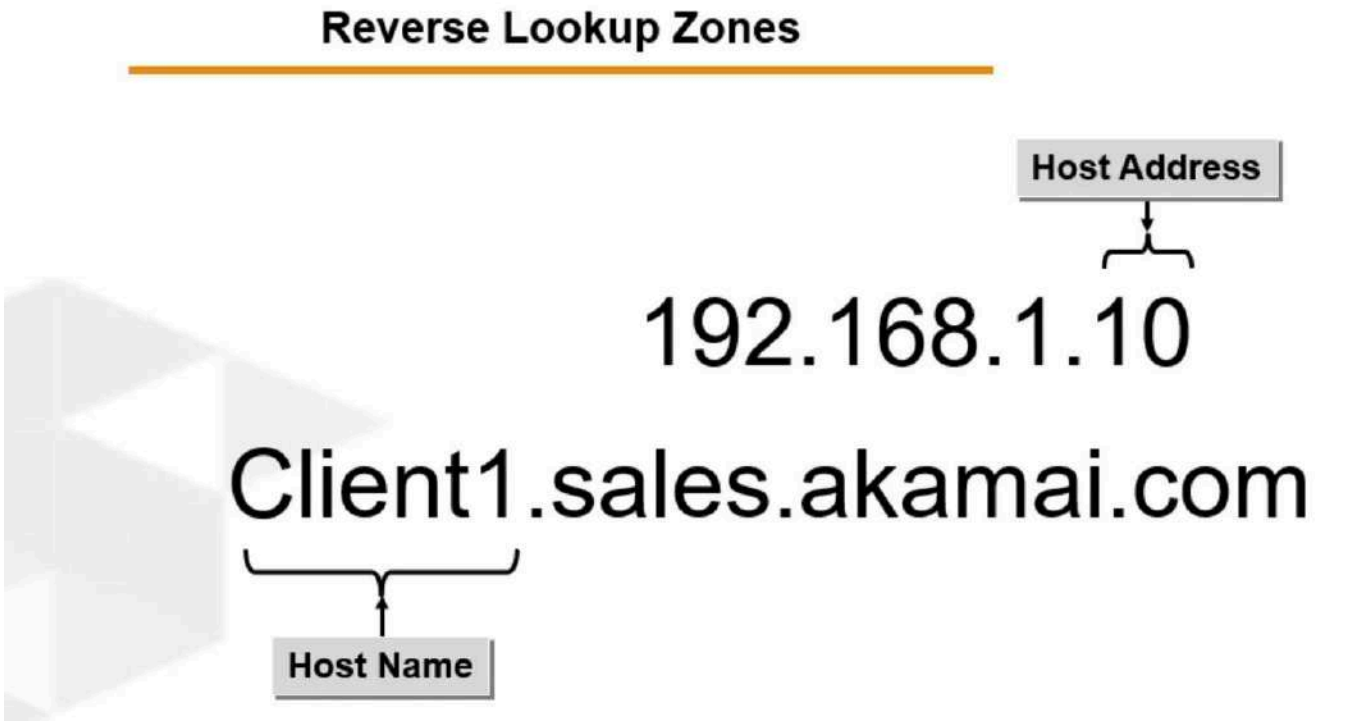
let's say that the dot com domain was hosted by a DNS server named DNS.com, and that, that server's IP address was this 34.56.78.90. That would be the delegation. Hey, I don't know about www. Akamai.com, but I know about.com. You need to talk to a server named DNS.com, and by the way, here's its IP address so that you know how to contact it. That information is going to come back to my DNS server. Now it's on the hunt. It's like a Sherlock Holmes or any type of detective that you like. It's going to find the answer. Now it's like, all right, let me just talk to the.com server. It contexts the.com server. Now if you notice we're moving left in that fully qualified domain name. We were at the period at the end now we're at.com, and the.com server is going to say, well, I don't know about www. Akamai.com, but I do know about akamai.com, let's say just for example, maybe it was a server named DNS that akamai.com. Oh, and by the way, that server's IP address is 12.34.56.78.. That information comes back to my DNS server. My DNS server now is going to contact the DNS server for akamai.com. We've moved left again in the fully qualified domain name, and akamai.com, that is the authoritative server for akamai.com. It's like www, I know that server. Here's its IP address. Let's pretend it's 56.12.34.56.78. That information comes back to my DNS server. My DNS server is going to put that in its cache in case anybody asks again, in the next 15 min to an hour, it sends that IP address back to the client. The client is going to put that in its cache in case it needs it again, and then the client is going to contact www.akamai.com via IP address. That's how DNS works. It's actually, I think it's really neat. Each DNS server is authoritative for just it's part of the DNS database, that's the distributed part, and the DNS servers use the fully qualified domain names starting at the root at the far right of the FQDN, and they work their way left to find the server that's authoritative, that's going to have the record that they need. That's it for this video. This was the second part of two parts on DNS name resolution. In this video, we watched the DNS server work through the DNS tree, starting with the root, navigating the different delegations until it could find the DNS server that was authoritative for the domain, and retrieve the IP address for the requested fully qualified domain name.

DNS Servers

DNS servers keep the DNS records in zones. And you can see there are two types of zones. Forward Lookup Zones contain records that take a name and match it to an IP address. You can probably guess that reverse lookup zones take an IP address and match it to a name, it's the reverse.



Let's take a look at how the reverse lookup zone works. We're looking at an IP address. We know that the network ID or the network address is on the left, and that the part that represents that particular node or host on the network is always on the right. So let's say I had this IP address 192.168.1.10, with the default /24 sub net mask. Then that .10 represents that particular host on the network. But when we look at a fully qualified domain name, the part that represents the actual host is on the left, right. So Client1.sales.akamai.com, Client1 is the host name.



So for whatever reason with the reverse lookup zones and the records, they said well, what would be nice is if we made it match. So they write everything backwards. So, if you had a record in reverse lookup zone matching the IP address 1921681.10, to the name Client1.sales.akamai.com. The record would be written like this, 10.1.168.192. And then all of the reverse lookup stuff ends with the .in-addr.arpa. So in address.arpa, you don't have to memorize all of that. I don't know if that's actually why they made it like this,

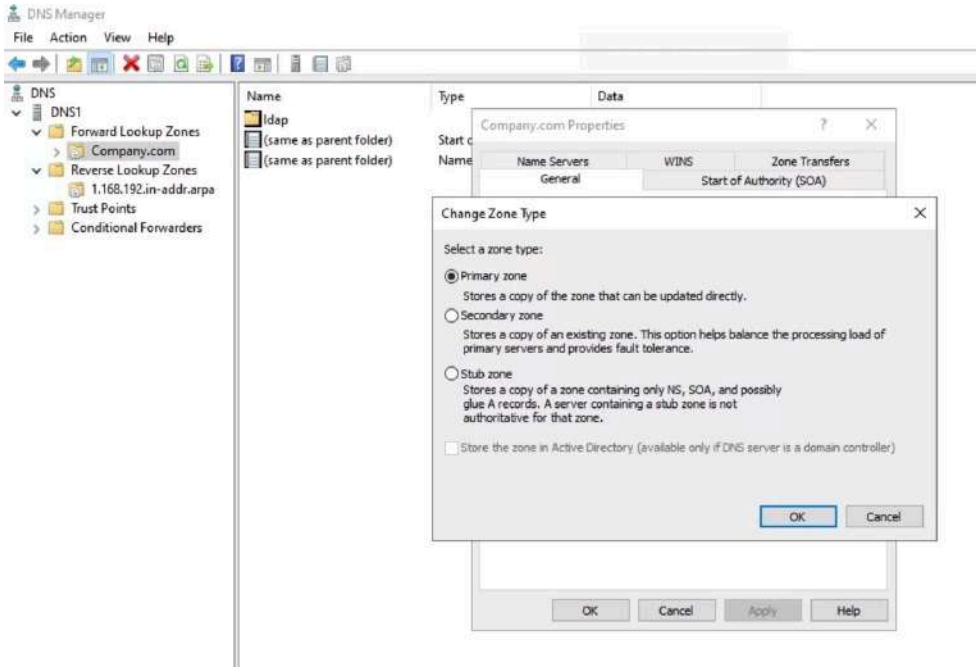
but it is true. Anything with reverse lookups, the IP addresses are written in reverse order. Once you know whether you have a forward lookup zone or a reverse lookup zone.

10.1.168.192.in-addr.arpa

Client1.sales.akamai.com

You have the actual zone files, and for each domain, there will only be one primary authoritative DNS server. The primary DNS server has the only read-write copy of the zone for that domain. That means only the primary server can accept changes to the DNS information in the zone. So if we go in here to company.com and look at the properties, you can see this is a primary copy of the zone, that means it's read-write. If I change it, my other option would be to be a secondary zone,

The secondary zone is a copy of the zone, and it's a read only copy. And you can see right here in this radio button it says it's going to help balance the processing load and provide fault tolerance. So, this read only copy at a secondary server would provide fault tolerance for that DNS zone. And fault tolerance means that if something goes wrong, and anything that goes wrong they say they call it a fault, then that means the service will not fail.

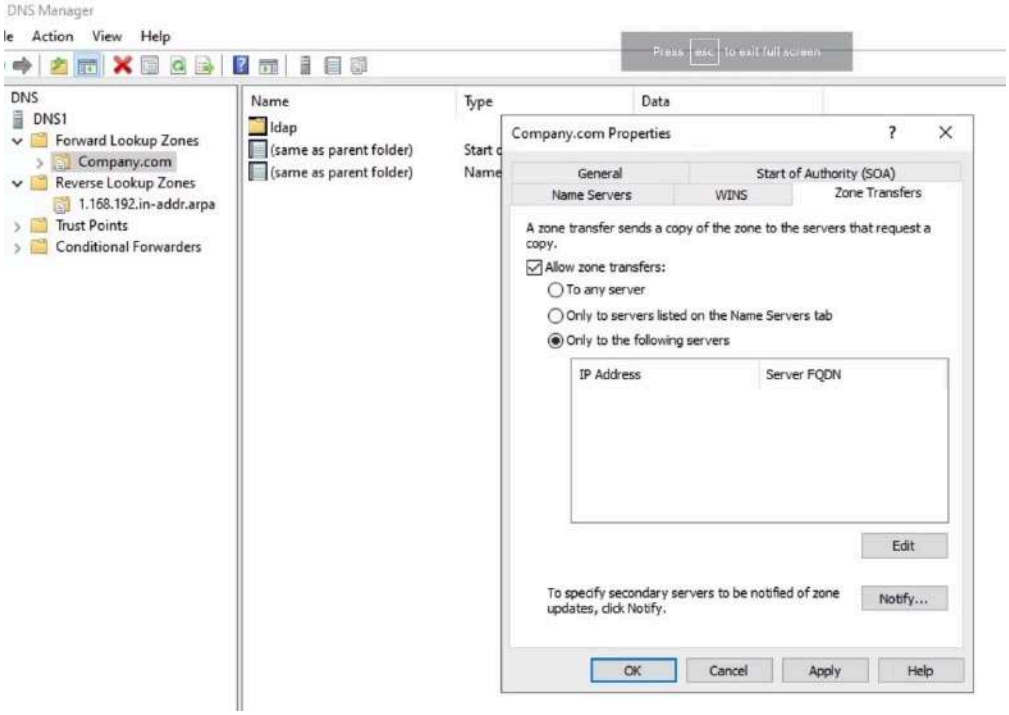


Now, people won't be able to make changes to the domain, because the primary copy of the zone is down. But they'll still be able to resolve names for that domain, because the secondary is up and running. Secondary servers can also help improve performance by providing additional servers to manage name resolution. If you think about it, DNS is super critical. Anytime you want to visit a spot on the Internet, you can't get there unless DNS happens first, because you probably start out with a name. What you actually need is the IP address. If the DNS server for domain is overburdened, then it's going to be slow providing name resolution. If it's slow providing name resolution, everything after that is going to be slow, and name resolution usually occurs first.

So the primary will conduct a zone transfer and give a copy of either the zone, if it's the first time or the updates if it's been a secondary for a while using a zone transfer. DNS zones contain all the records for a domain. Now it is important to limit the devices that can receive a zone transfer, to authorize devices to prevent hackers from gaining too much information.

DNS is tough to secure because it's a public database. The whole point is that anybody can ask what is the IP address for www.akamai.com and get an answer. But, still we at least want to make the hackers sit there and ask what's the IP address for this? What's the IP address for that. If they can get a zone transfer, they can get all the information in DNS very quickly. And if you think about it, usually that's going to be all the names of all your devices, all their IP addresses, that is a great place to start for information. And so typically with a zone, we would want to at the very least limit zone transfers to the secondary servers.

And in a Microsoft DNS server, it's actually in the properties of the zone, you have a zone transfer tab that says, "look, who can have a zone transfer". The very least we want only the servers on the name servers tab, it would just list the primary server and the secondary server. So, effectively that middle radio button says only two secondary servers. Or if I'm concerned about somebody impersonating a secondary server, I could say look, only to the servers that have this particular IP address and list them out.

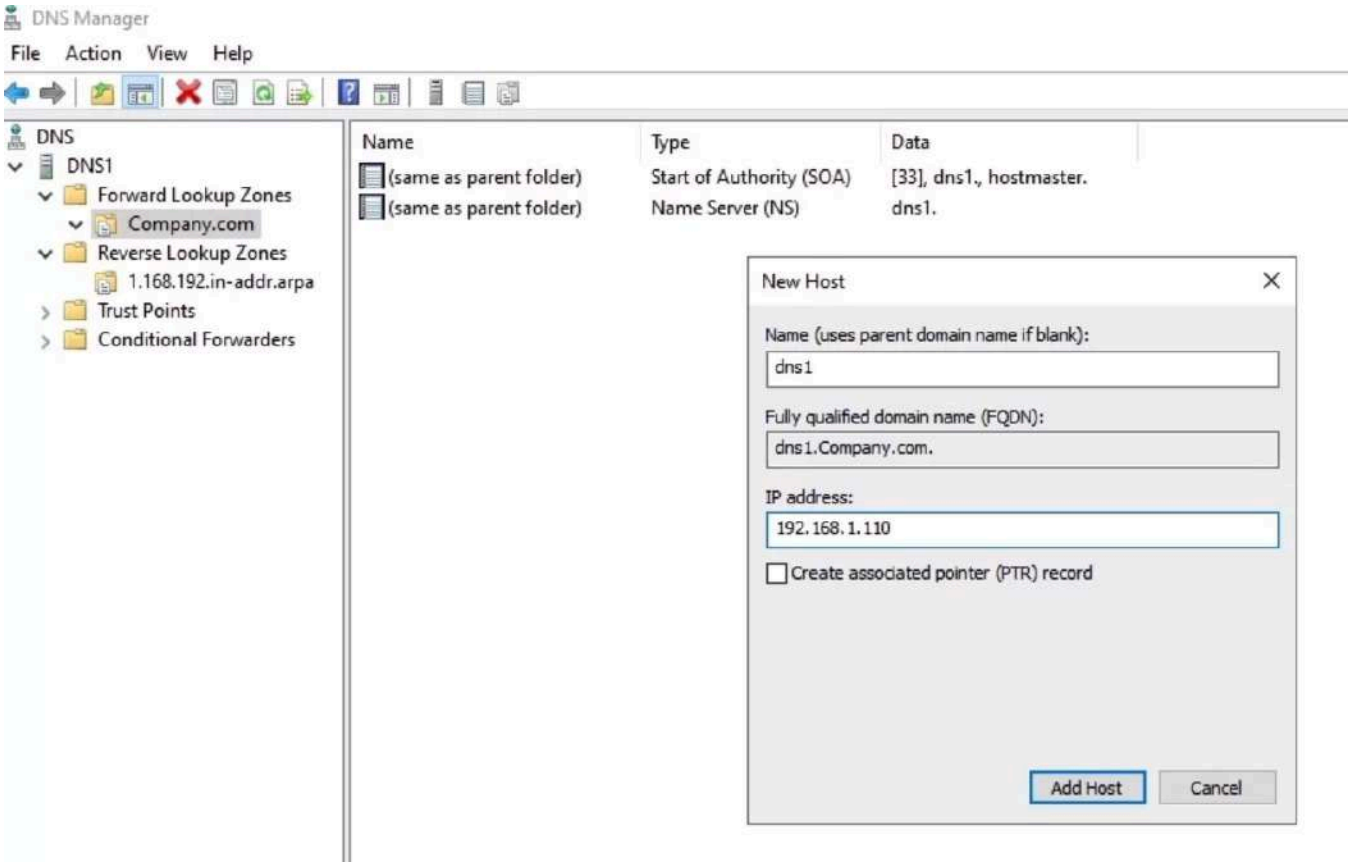


But we do want to limit who can have a zone transfer for security purposes. In the zones we can have static records that are created by the administrator, or we can have dynamic records that are created by clients checking in.

we've taken a look at the DNS server. We've seen Forward Lookup Zones, Reverse Lookup Zones. Forward Lookup Zones are named IP, Reverse Lookups are IP to name. We talked a little bit about this naming structure for reverse lookup zones. We talked about primary DNS servers which have the only rewrite copy of his own. Secondary servers that have read only copies for fault tolerance and load balancing. And the process of zone transfers where they primary will send changes over to the secondary.

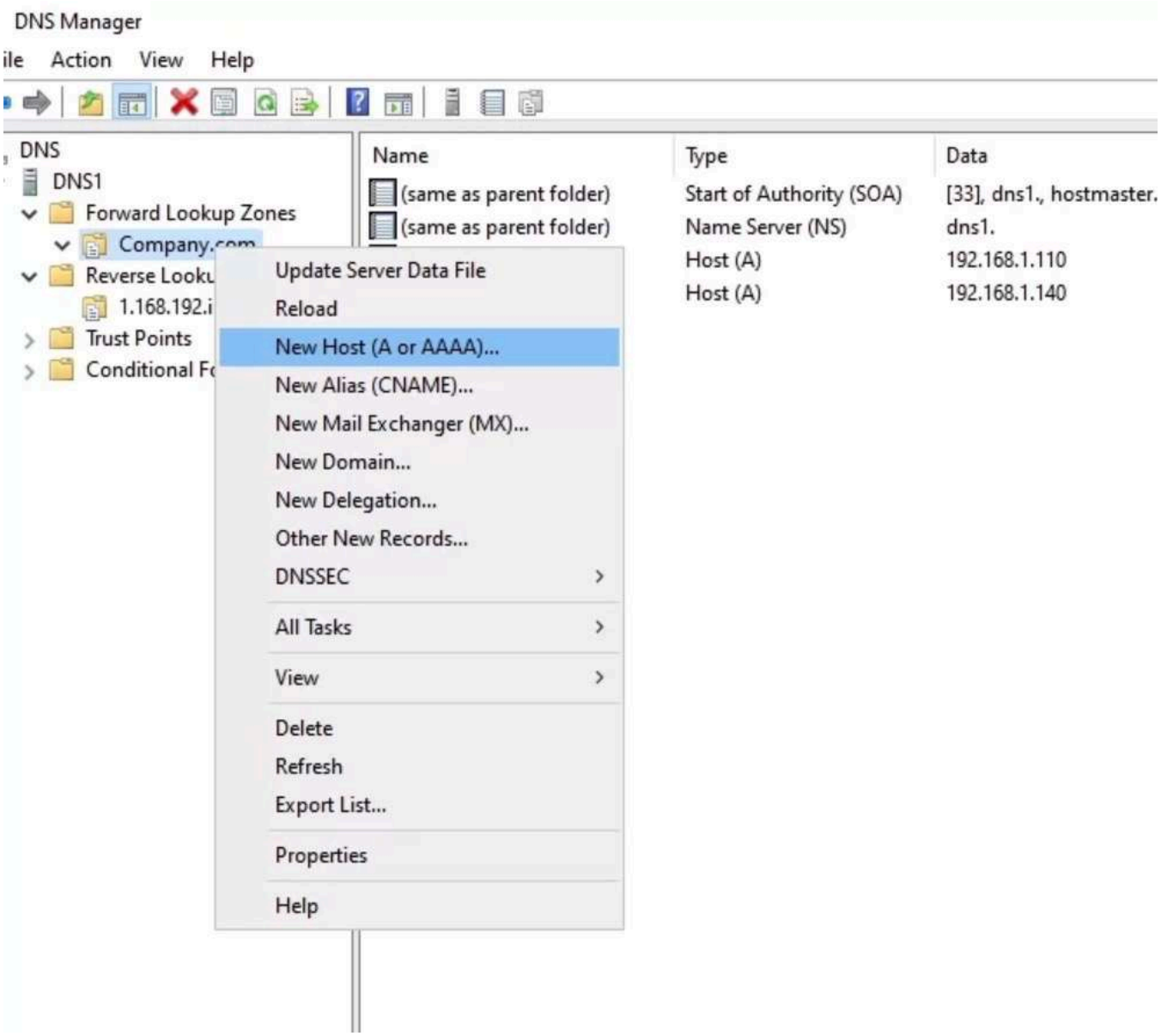
DNS Records Part 1

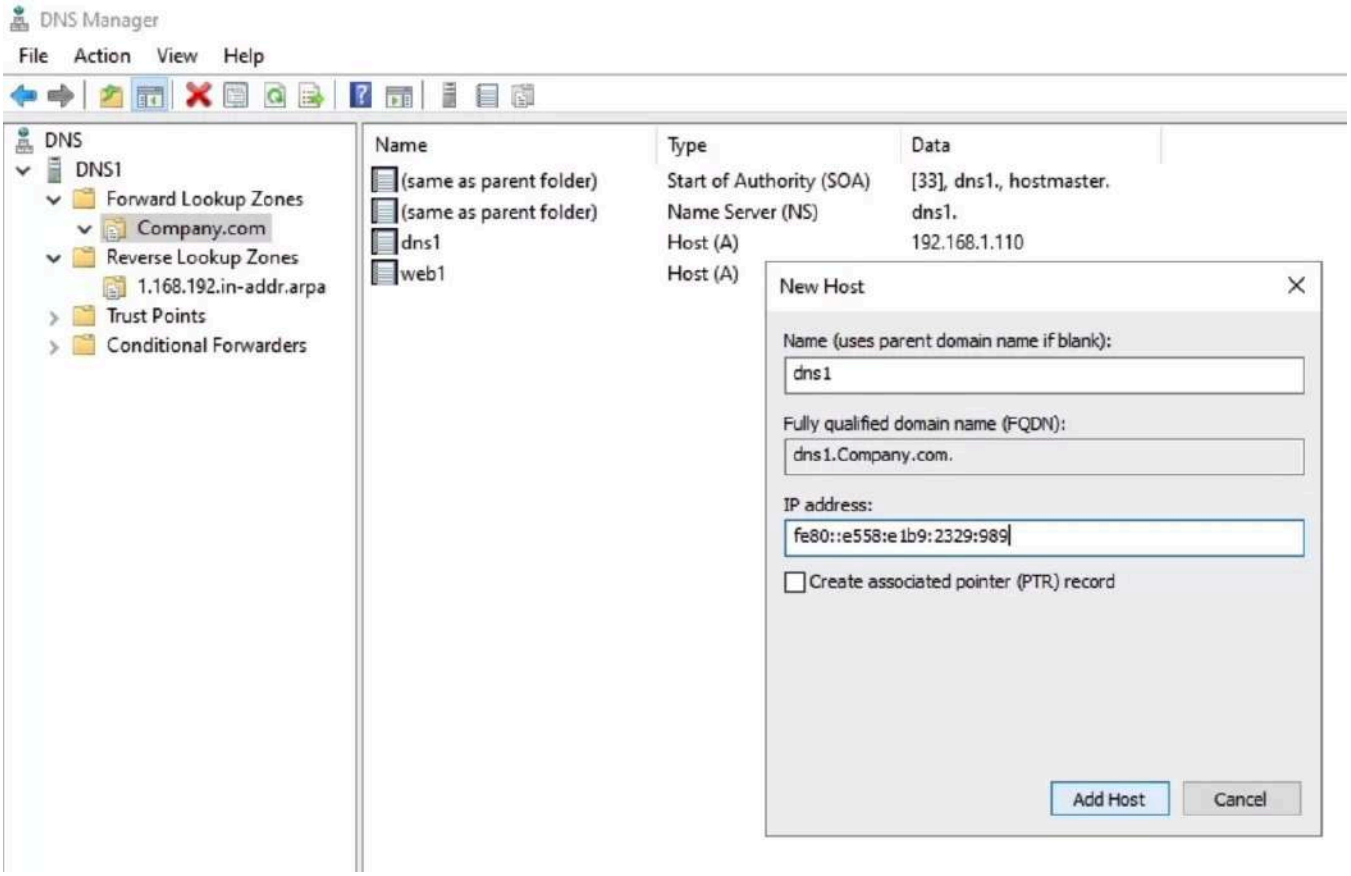
The first type of record we want to look at is an A record. I believe the A stands for address, but honestly, nobody ever says that. They just say A record. I'm here in my Microsoft DNS server, and they actually call it a host record. I'm going to right-click my "Company.com" zone and make a new host record. An A record maps a hostname to an IPv4 address. Here, I'm actually going to go ahead and put in the name of my DNS server, which is dns1. Then it's IPv4 address, 192.168.1.110.



You can see there's my A record. When you create the A record, it won't be an issue in the labs for this course because you're going to be working in packet tracer. But if you're ever working in a DNS server, notice you just type the name of the host up here. You don't type web1.company.com

Now IPv4 IP addresses are 32-bit numbers. We also have records for IPv6 addresses. IPv6 addresses are 128-bit numbers. I would have to assume that somebody said, wait a minute if we have an A record for an IPv4 address and IPv6 addresses are four times as big, why don't we make the IPv6 records a quad A record? The next record we're going to make is a host record, but we're going to be making a quad A, and by that, I mean four A's like AAAA. That's the record. We're going to go ahead and do a record for DNS1, but I'm going to need its IPv6 address.





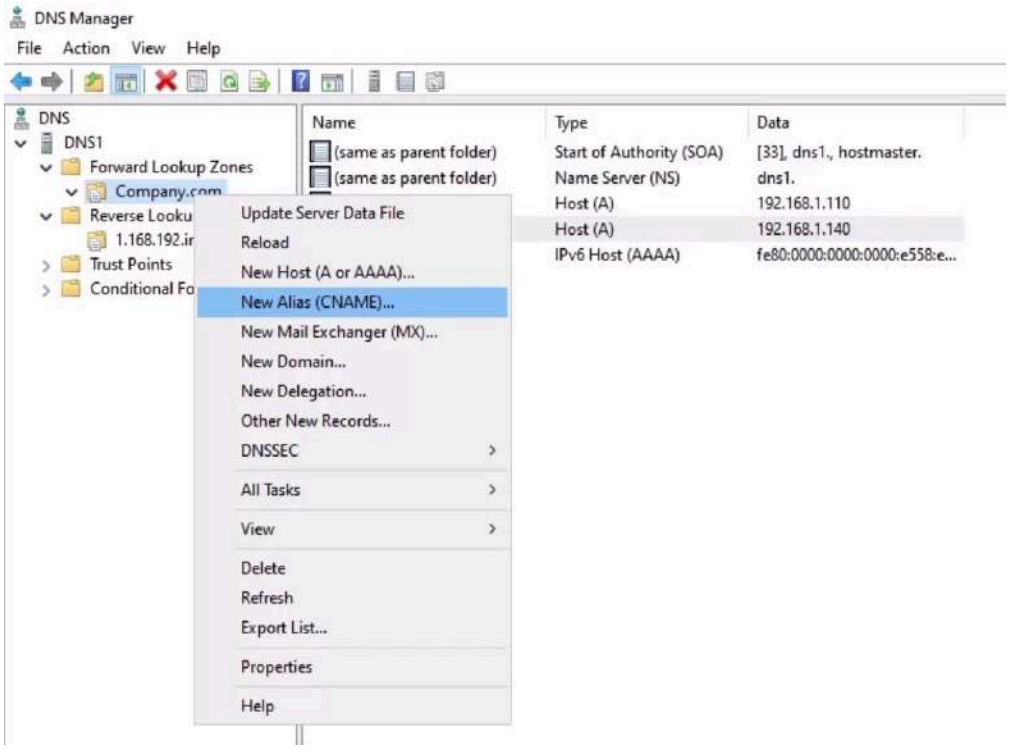
Now another type of record that we often make is a canonical name or CNAME record. Nobody says canonical name.

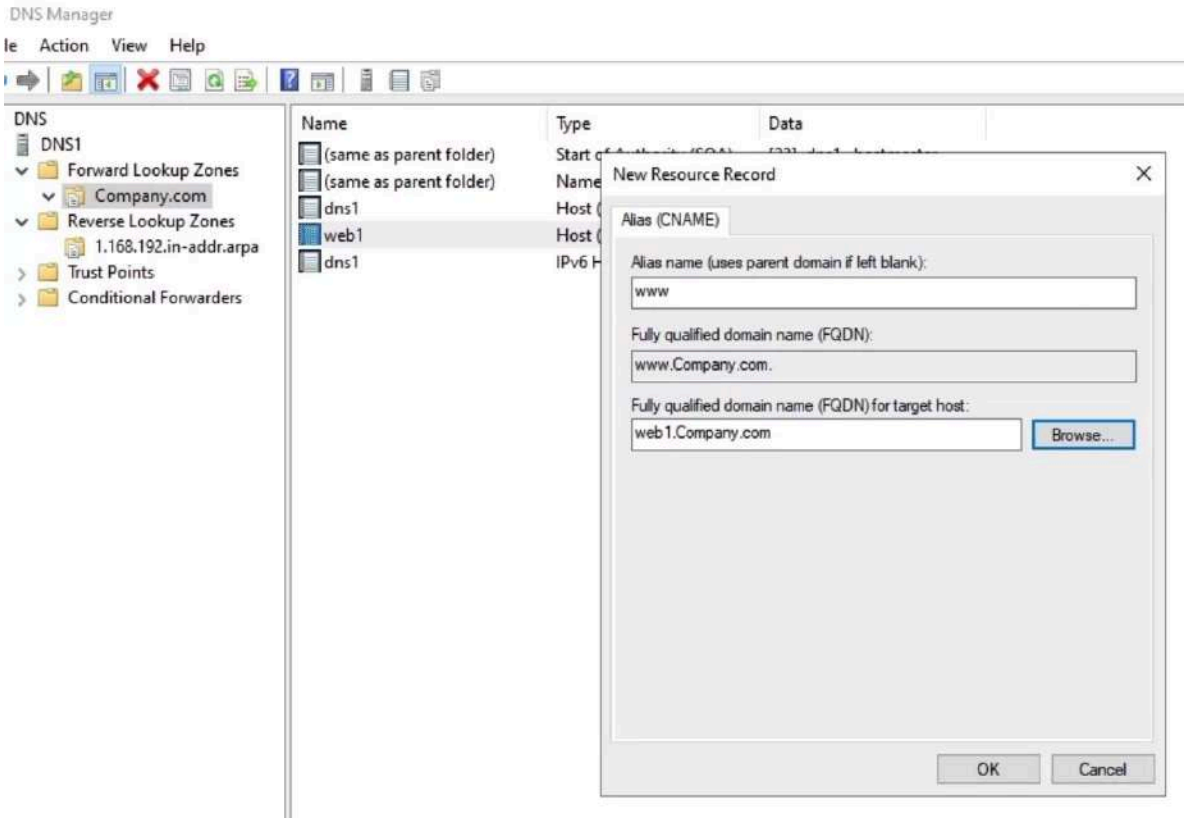
CNAME record is an alias record. It's used when you want to provide another name for a server. Let's say that this web1.company.com is hosting the company's website at www.company.com, but web1 is the actual name of the server. I could just create another A record. Www is the same IP address, but here's the problem. If that server changes its IP address, it's going to check in with DNS and say, hey, I've changed my IP address, I need to update the record, and it's only going to update the web1 record. In DNS, there's two types of records. There's static records, which is what we've been doing where they're created by the administrator. Those records would not change unless an administrator updates them. But there are also dynamic records that are created by the host themselves when they check in with the DNS server to report, hey, this is my name, this is my IP address. We don't want to create another A record for www because in that case, it'll be a static record. It's not going to be updated if web1 changes IP address. We can avoid a human being having to be involved by creating a CNAME record.

Let's go ahead and make a CNAME record. You can see Microsoft calls an alias. Here up top I put the alias in, so this is going to be www.company.com.

Then I need to tell it the A record of the server that this is an alias for. We're going to get that web1 record, and there's my CNAME record. CNAME is just an alias. It's another name. You're like, my name is Shadow

Feral, but my alias is Shad. It's like that. Another name for a server, and they're very commonly used, particularly when you want to host multiple websites on the same server.



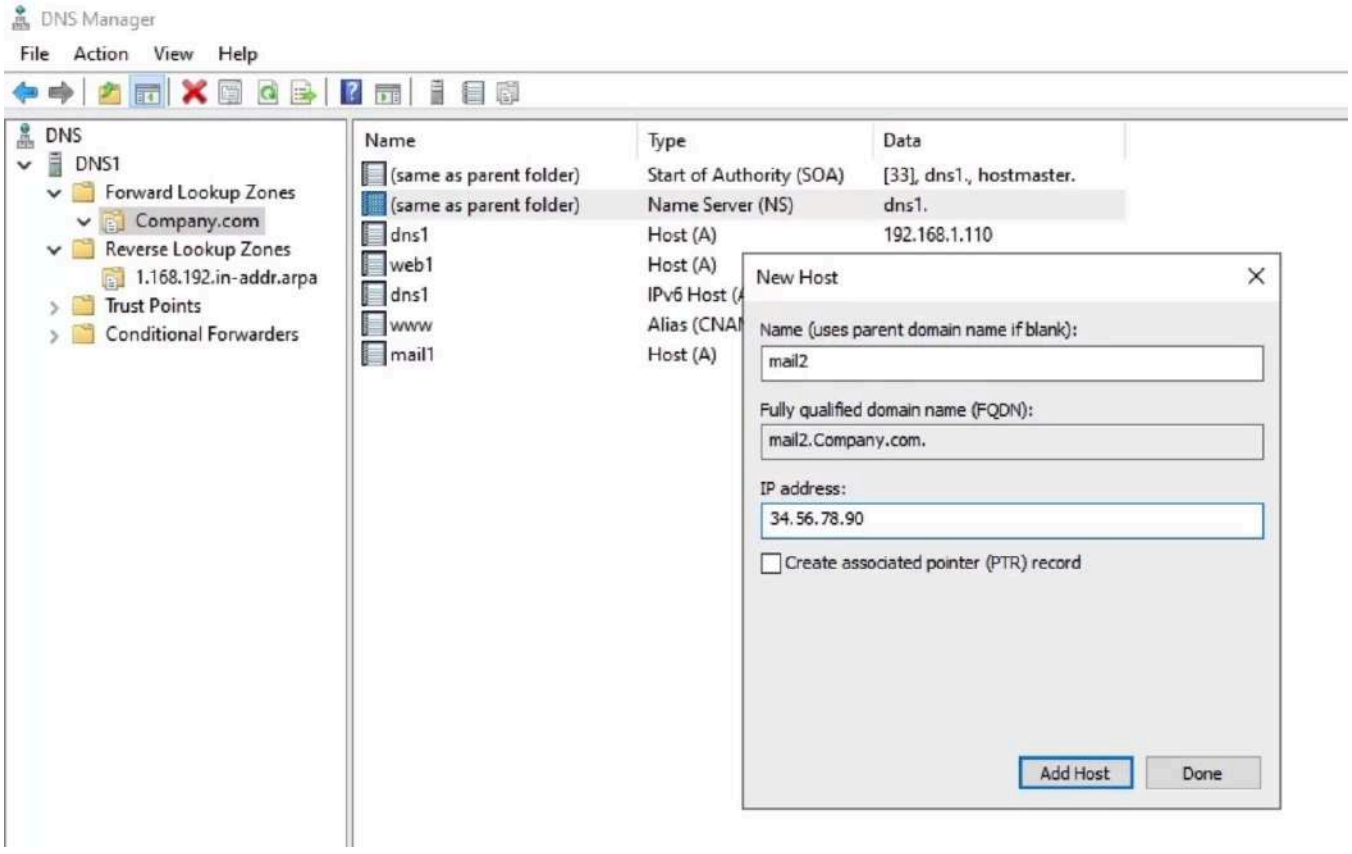


Another type of record that you would see would be an NS record or a name server record. Name server records identify DNS servers. When you see an NS record, it's saying this is the DNS server for this particular domain. The only thing I found confusing about that when I was starting out is you would think it would be a DNS record, but it's not. It's an NS record that identifies the authoritative DNS server for the zone.

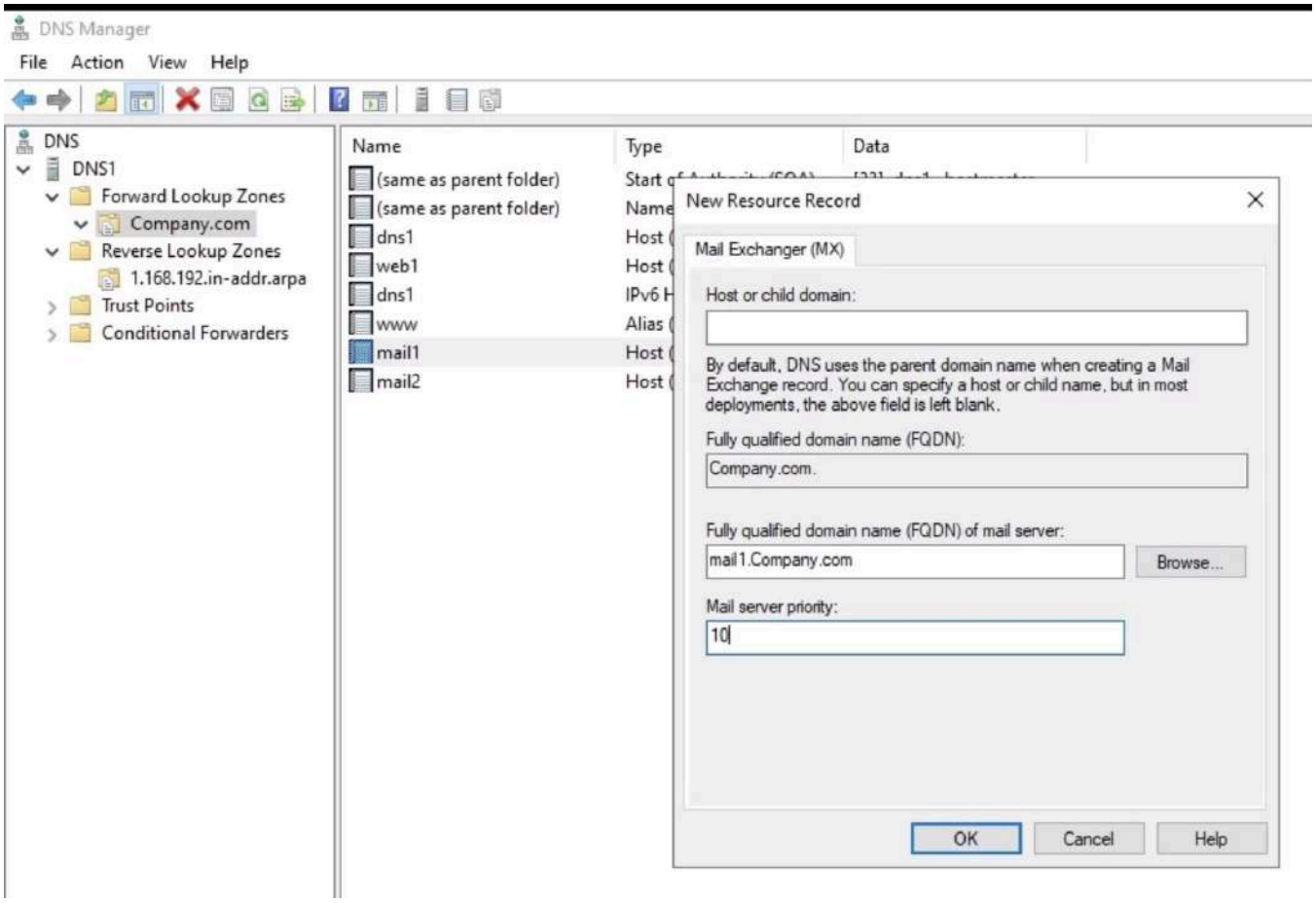
We looked at the A record which matches a hostname to an IPv4 address, the quad A record that matches a hostname to an IPv6 address, CNAME record, which provides aliases for devices, meaning another name that can be known by; very commonly used when a server host multiple websites, and NS records that identify authoritative name servers, meaning DNS servers for that particular domain.

DNS Records Part 2

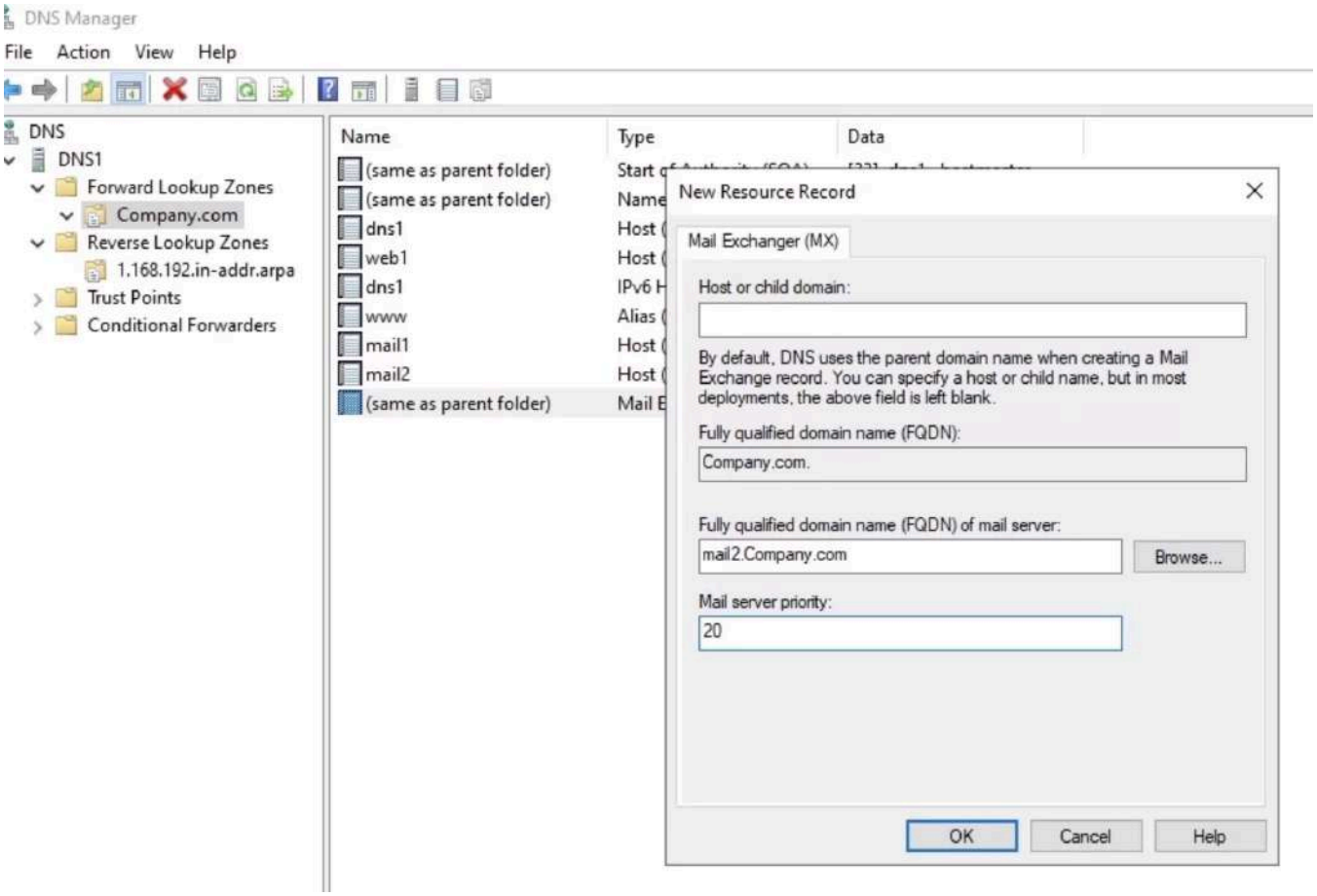
The first record we want to look at in this video will be MX records or mail exchange records. They map a domain name. They identify the server that's handling mail or email for a particular domain, and they point to the A record of the server that's going to handle email. Let's go ahead and create a couple of A records for our email servers. Let's say one is named mail1.company.com. We'll just give it a fake IP address here. Then we'll make another one, mail2.

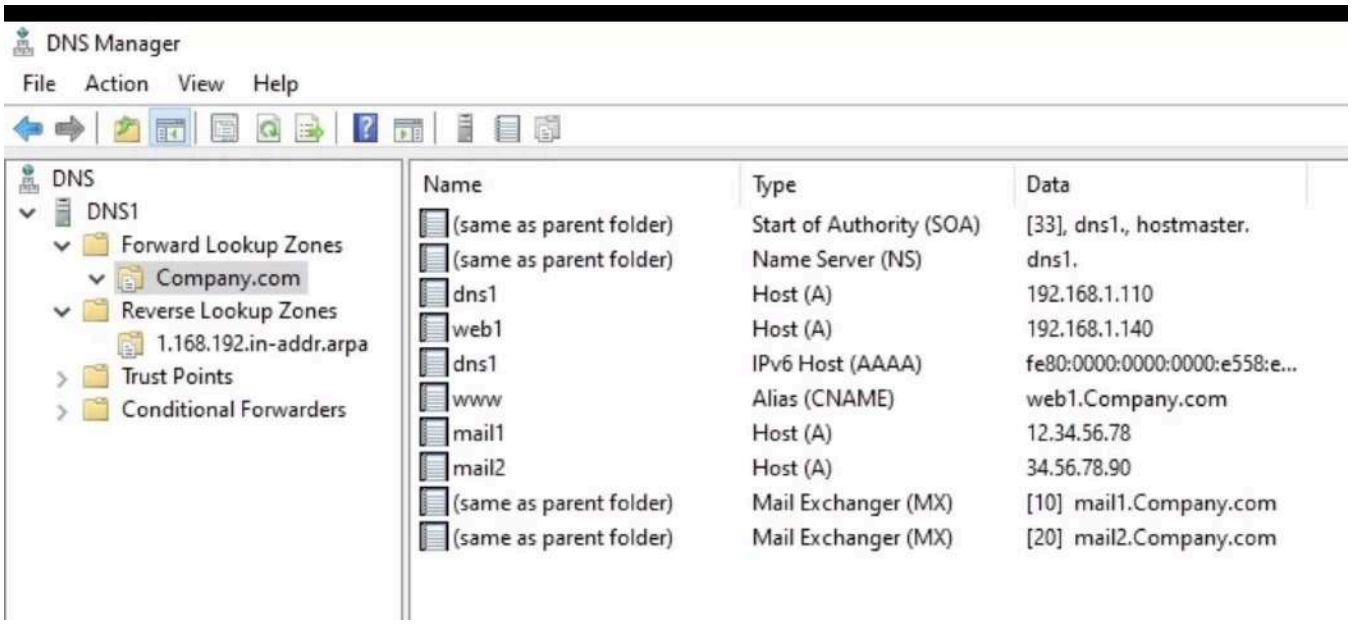


Just because I named the servers mail1 and mail2, that doesn't tell anybody that these are the servers that handle email. For that we need MX records. I'm going to go ahead and create a mail exchanger record, MX record. By default, it's for the domain that you're creating it in, and that's what it's telling us up here in this first paragraph. If it was for sub-domain, you could put something in here. I'm going to leave it as is, and it just says, okay. If the email is going to company.com, then this is going to be the server that you need to contact. Then I need to specify the A record for the server that other mail servers or client should contact. I'm going to go get the record for mail1.



Now, MX records are a little bit different in that they have a mail server priority, where the lower the number, the higher the priority. The default is 10. It really doesn't matter. You could use one, you could use 100, you could use 1,000. The numbers don't mean anything except when they are compared to each other. We're actually going to make two MX records. I'm going to leave mail1 at a priority of 10, and you can see that this new record is down here. Then let's make another MX record that's going to point to mail2.

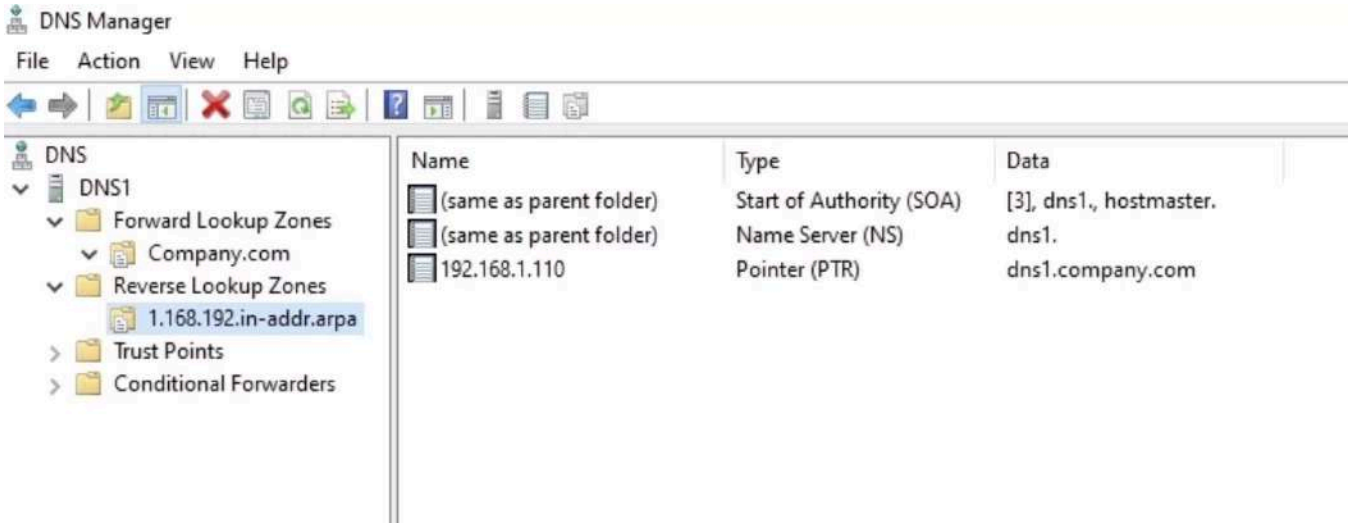




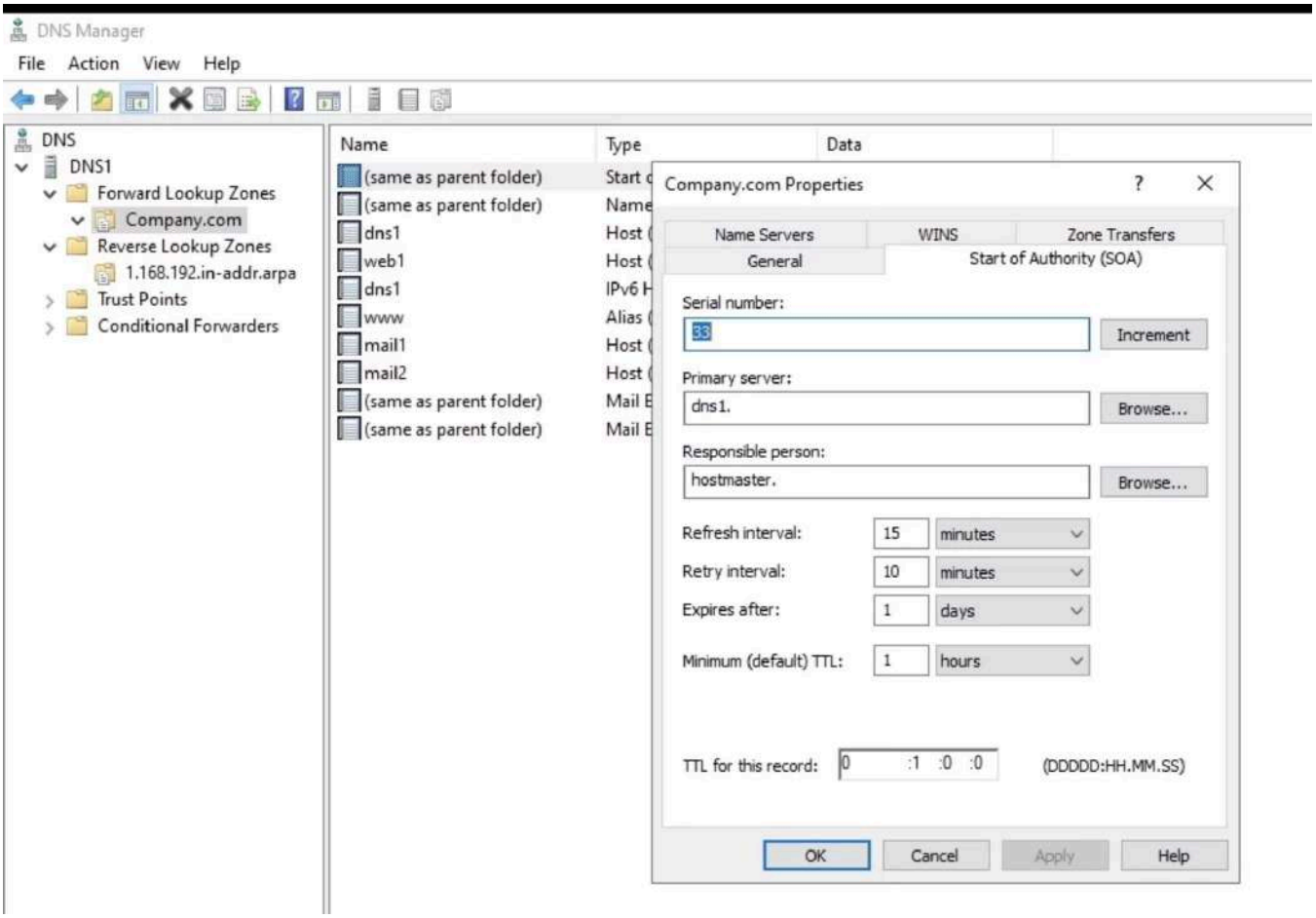
But let's go ahead and give mail2 a priority of 20. It doesn't matter. It could be 11, anything more than 10. When clients check in, let's say it's another email server that wants to send an email to company.com, it

would request all the MX records for company.com and then both of these two records here would come back. It will always choose the MX record with the lower priority unless it's not available. The way I have this setup right now, every server that wants to send email or read email, if it's a client, they're going to read the mail, they're always going to use mail1.company.com. The only way it would go to mail2 is if mail1 is not available. If there are multiple servers that have the same priority, let's say both of these had a priority of 10, then it will just split the work both ways. By assigning different priorities, I'm essentially setting a backup mail server. If I assigned the same priority, I'd be splitting the work. But mail servers do have a priority. The lower the number, the higher the priority, or the more preferred server, that is

The next record we're going to look at is a Pointer record. A records match a host name to an IPV4 address, AAAA, host name to IPV6. Pointer records are the opposite of the A or AAAA record. They map an IP address to a name. I'm going to go ahead and make a Pointer record and say, okay, if you have 192.168.1.110, then that's dns1.company.com. I can either type it in or I could go browse for the record. The Pointer record would be the same whether it's IPV4 or IPV6. In this case, it knows it's IPV4 because I'm in an IPV4 reverse look up zone. But if I had an IPV6 reverse look up zone, it would still be a Pointer record if it was an IPV6 address to a name. Pointer records, they're the reverse of the host records. They map IP addresses to name.



Another record we want to take a look at is an SOA, or start of authority record. You can see one here in my reverse lookup zone, is going to be wanting company.com as well. This is the first record in any zone, and it contains information about the zone. Microsoft has integrated this into the software. When I double-click the SOA record, it's going to pop up with the properties of the zone.



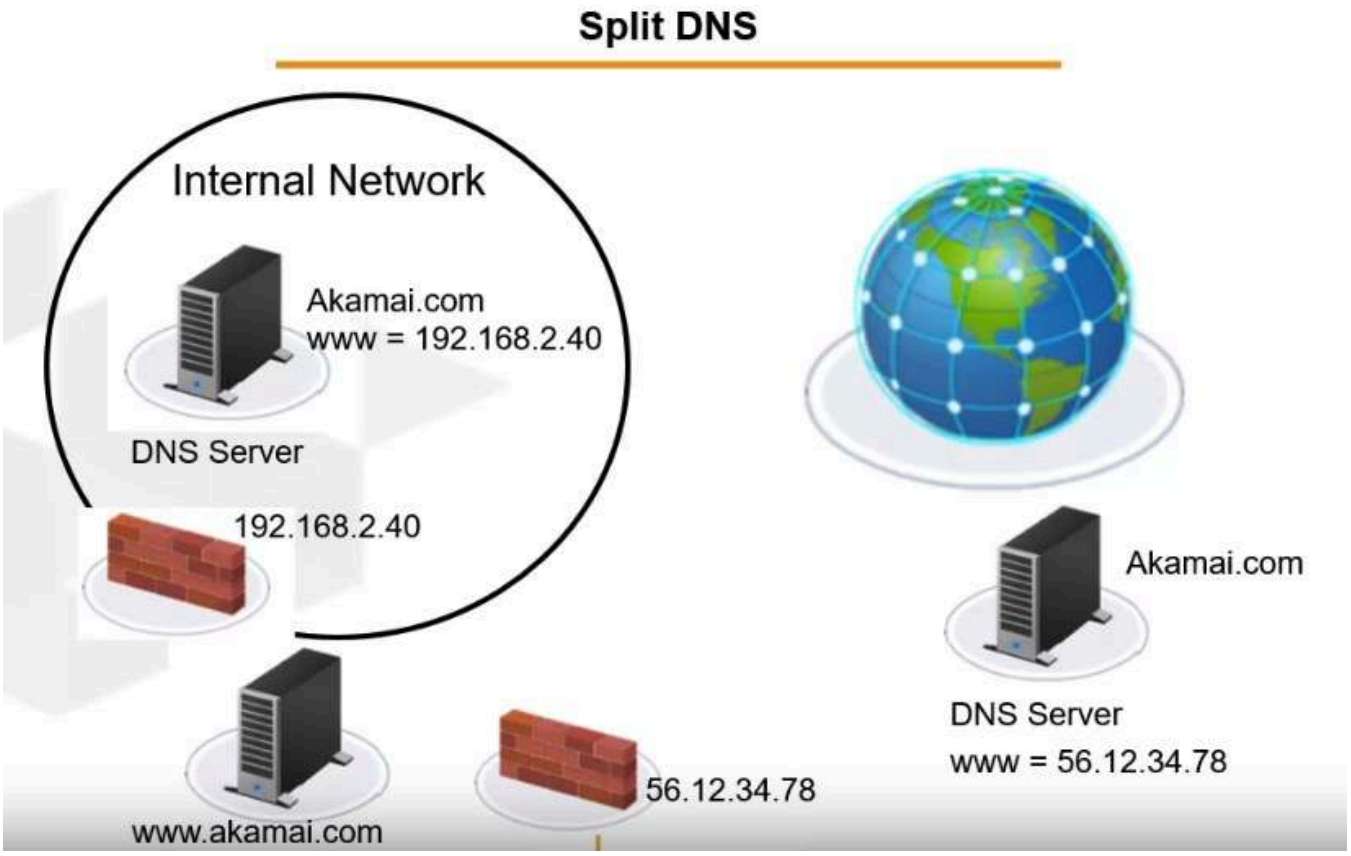
But even in non-Microsoft zones, there's always an SOA record. The SOA record has a serial number that gets incremented every time you make a change to the zone. This is how the secondary servers know when they need a zone transfer. Let's say right now my SOA record is at 33, I'm on the primary server. We would assume that the secondary servers are also at 33. If I made seven changes, that serial number go up to 40. Every so often, the secondary server is checking with the primary and say, hey, give me your SOA record. If they see that the current serial number on the primary is different than the SOA record they have, they know changes have occurred, and they ask for a zone transfer.

The SOA record identifies the primary authoritative server, which we know there's only one, that's the one with the read-write copy of the zone, and then responsible person is the email address of whoever manages this domain. They use a dot instead of an @ symbol. What this is telling me is that if I want to contact the administrator for this domain, I would email hostmaster@company.com.

The time to lives are how long records from this domain will stay in a DNS cache after it's been resolved by another DNS server. Then the refresh interval is how often the secondary would check in to see if it needs a zone transfer. Retry would be how many minutes go by if the zone transfer fails before it should retry again. Expires after is how long will it keep answering people's queries for records from this zone without a zone transfer. You do not need to memorize any of this. If you know the SOA record has information about the zone, I think you're going to be in great shape.

In this video, we finished our discussion of DNS records. We talked about the MX records which identify email servers for the domain. They point to the A records of the servers that are in charge of mail. We looked at Pointer records which map IP addresses to names, and we took a look at the SOA records, start of authority record, that has information about the zones.

Split DNS



DNS is difficult to secure because DNS information is intended to be given out to requesting servers. Pretty much anybody who asks, we want to give them the information. But for security, many companies implement Split DNS. Let's take a look at the scenario and I think you'll understand right away why this is great. Let's pretend that Akamai has just one internal network, and in here, akamai.com, there would be records about all of the devices at Akamai, all the clients, all the servers, everything. But all of those things are private, they are just for internal devices of Akamai. Let's pretend there's only one server at Akamai that should be accessible from the Internet, and that's this server over here, www.akamai.com.

For the sake of example, this server is going to be located in a demilitarized zone, which means there's a firewall between the server and the Internet, and there's a firewall between that server and the internal network. Now, we need to provide name resolution for www.akamai.com so that people on the Internet can get to that website. But we wouldn't want to have a DNS server out here on the Internet that has a secondary zone for akamai.com because a secondary zone is a read-only copy of the primary zone. That would mean that all the records about all the internal devices at Akamai would be sitting out on the Internet for anybody to request that information.

With Split DNS, there are two servers that are authoritative for a zone. Each of these servers, the DNS server on the internal network and the DNS server on the Internet, they would each be primary authoritative servers for the akamai.com zone, but they would have different records. The internal DNS server would have all of those records. For everything at Akamai, the external DNS server out on the Internet would just have the one record for www.akamai.com.

That way if somebody is trying to use DNS to research Akamai, that's all they can find out about. The only thing in the primary DNS server on the Internet would be records for public resources. Whenever we say a resource is public, we mean accessible by anybody who's out on the Internet. Now what does get interesting here is that they would each have a record for www.akamai.com, but they would have different IP addresses. Let's say that this 192.168.2.40 address, that's the internal IP address of this firewall that's between the internal network and the demilitarized zone where this server sits.

Well, that's what the internal record would have to say. If you're looking for www.akamai.com, you need to talk to this firewall and then the firewall is going to pass you along to the web server. For people out on the Internet, let's say that this external firewall, its IP address on the Internet is 56.12.34.78. That would be who Internet clients would have to be sent to.

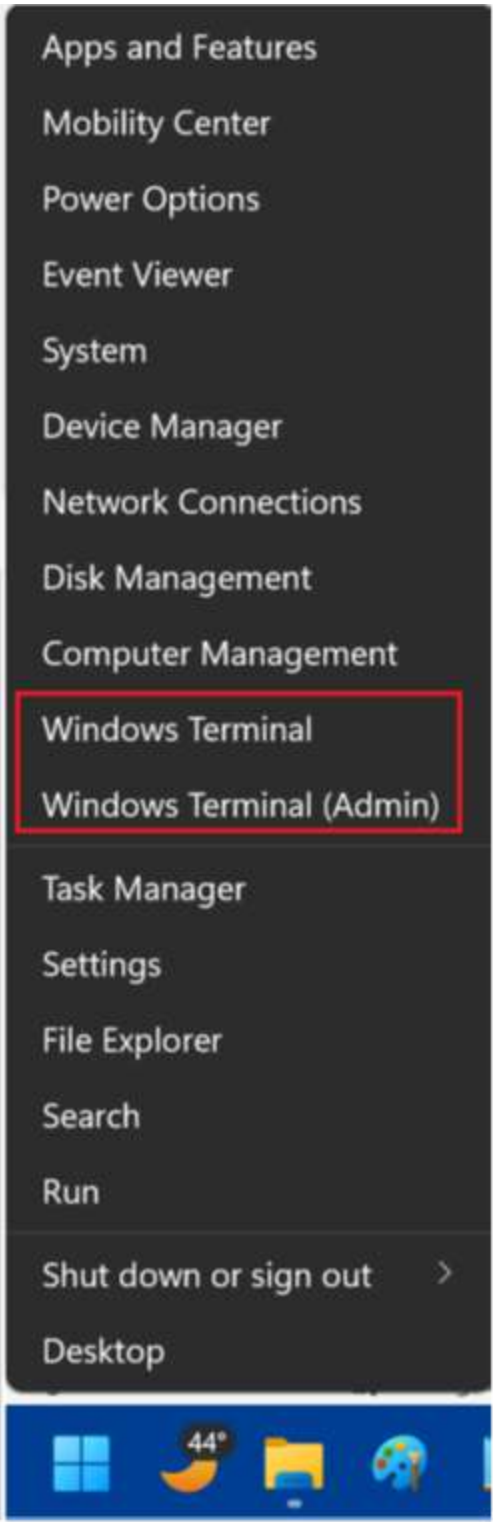
They would need to be sent to talk to this firewall, and then that firewall would pass it along to the server on the other side of the firewall. Split DNS it's a way of protecting DNS. It means that there are two primary authoritative DNS servers for a particular domain. One is an internal server that has all the records, the other is an external server out on the Internet, and it contains only the records for the public resources.

TCP/IP Commands

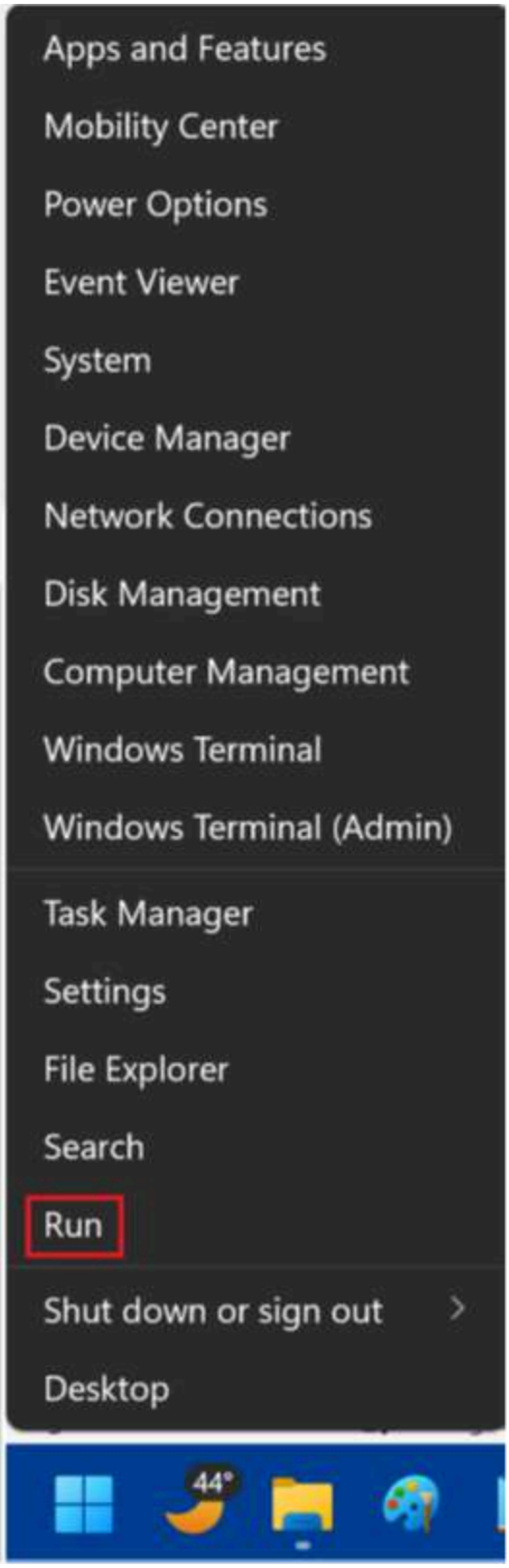
IPConfig/IFConfig
When you're troubleshooting connectivity, the first step is to determine the IP address is of the client you're working on. This can be done inside the graphical user interface but how is dependent on what operating system. It can also be confusing if the device has multiple network cards.
The easiest and best way to find out the IP address is to use a command prompt. On some operating systems the command prompt may be called a "terminal."

Ipconfig

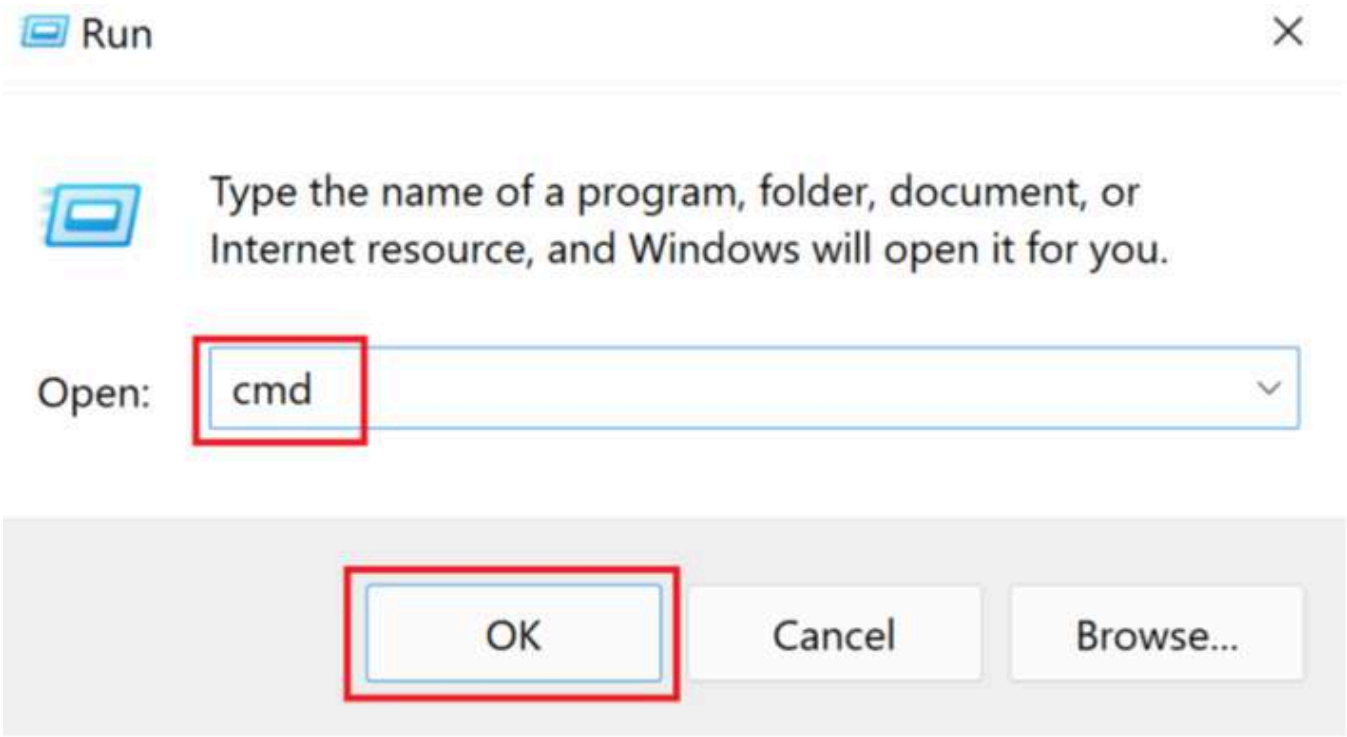
In a Microsoft Windows ® operating system, you can usually get into a command prompt or terminal by right clicking the Windows button on the taskbar.



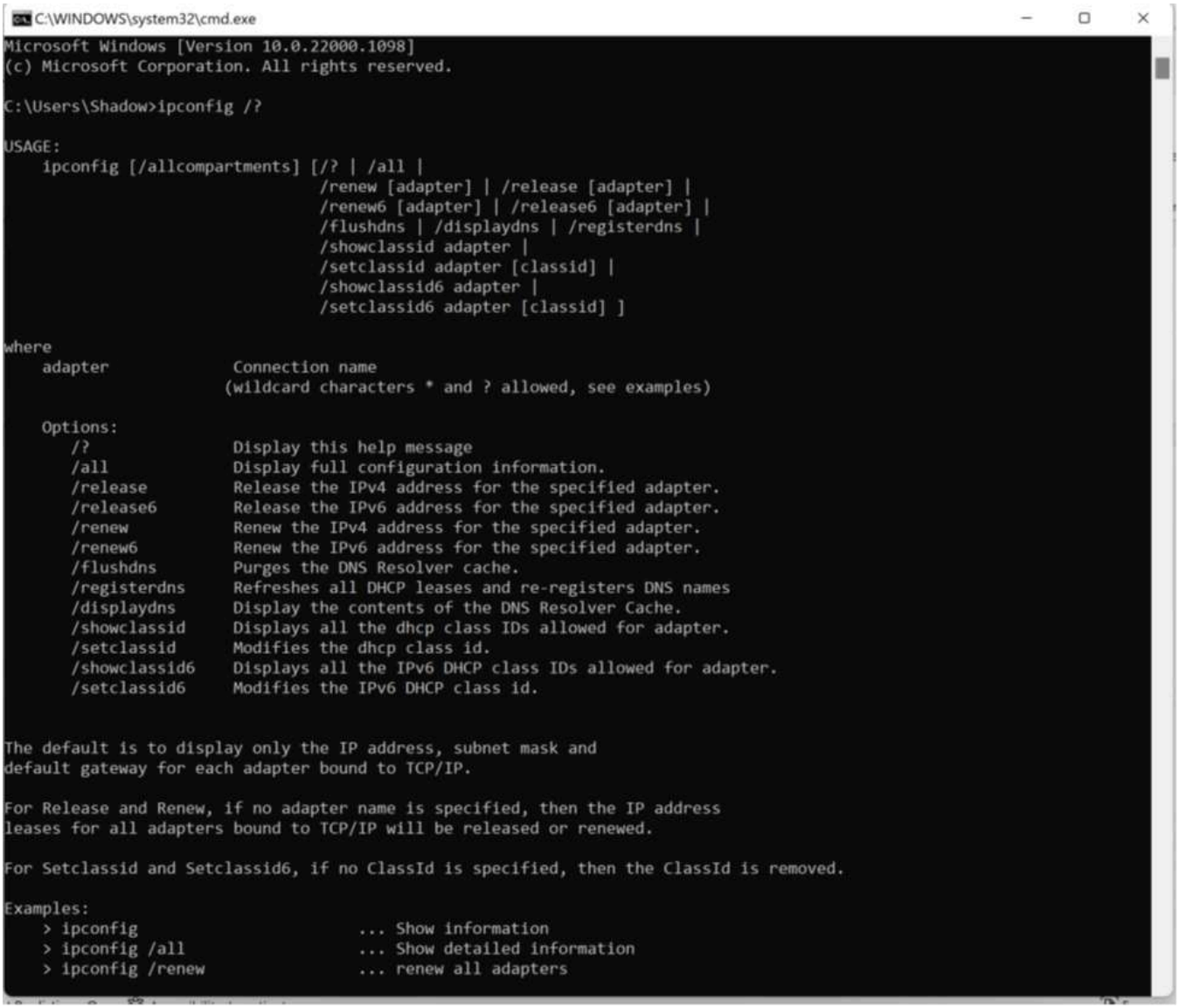
If you don’t see “Command Prompt” or “Windows Terminal” on the menu, click **Run**. (Note: ipconfig also works in the PowerShell prompt.)



In the **Open** text box, type **cmd** and then click **OK**.



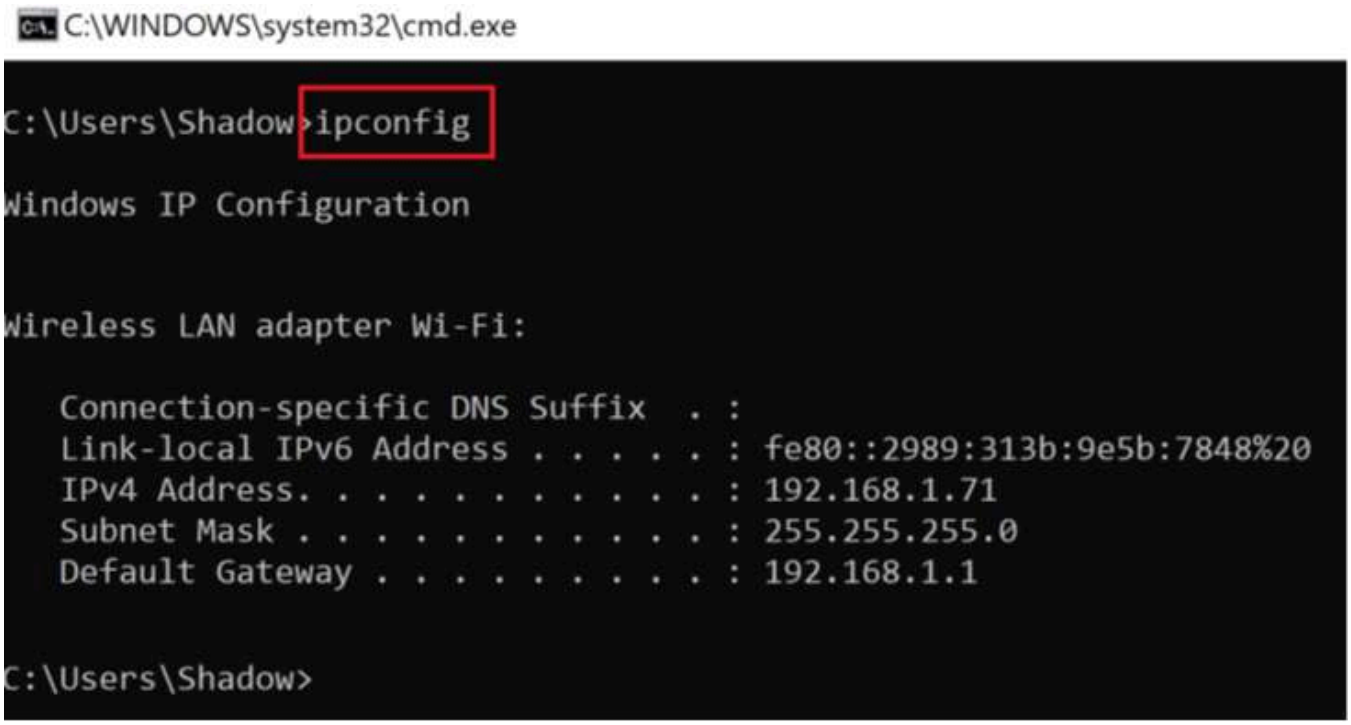
In Windows, the command to work with the IP address is ipconfig. Ipconfig has a lot of different options. You can see all the ipconfig switches by typing **ipconfig /?**



The most common ipconfig switches you will use are:

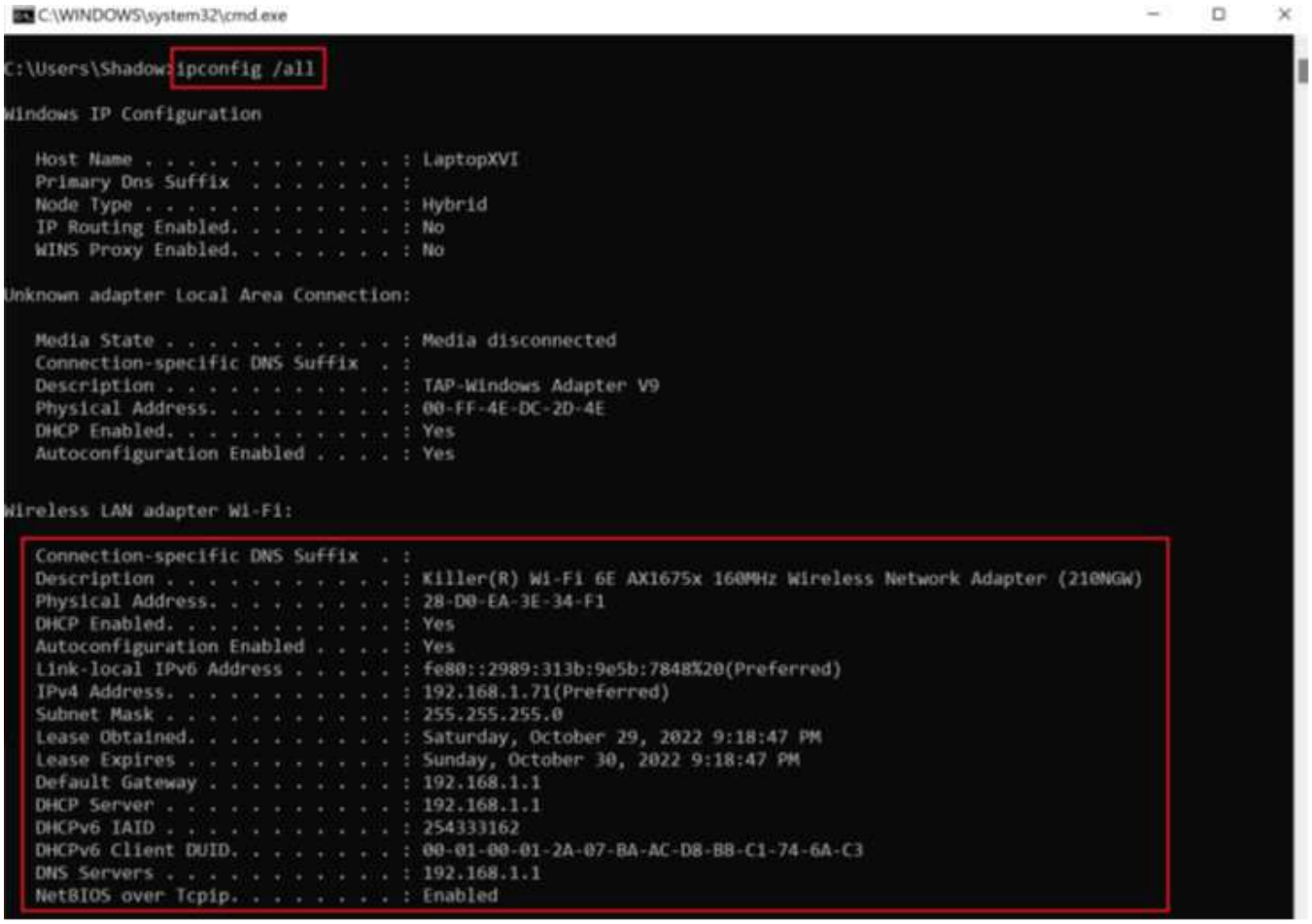
Ipconfig

Shows you a small amount of information about all the interfaces in the computer



Ipconfig /all

Shows detailed information about all the interfaces in the computer.



Ipconfig /release

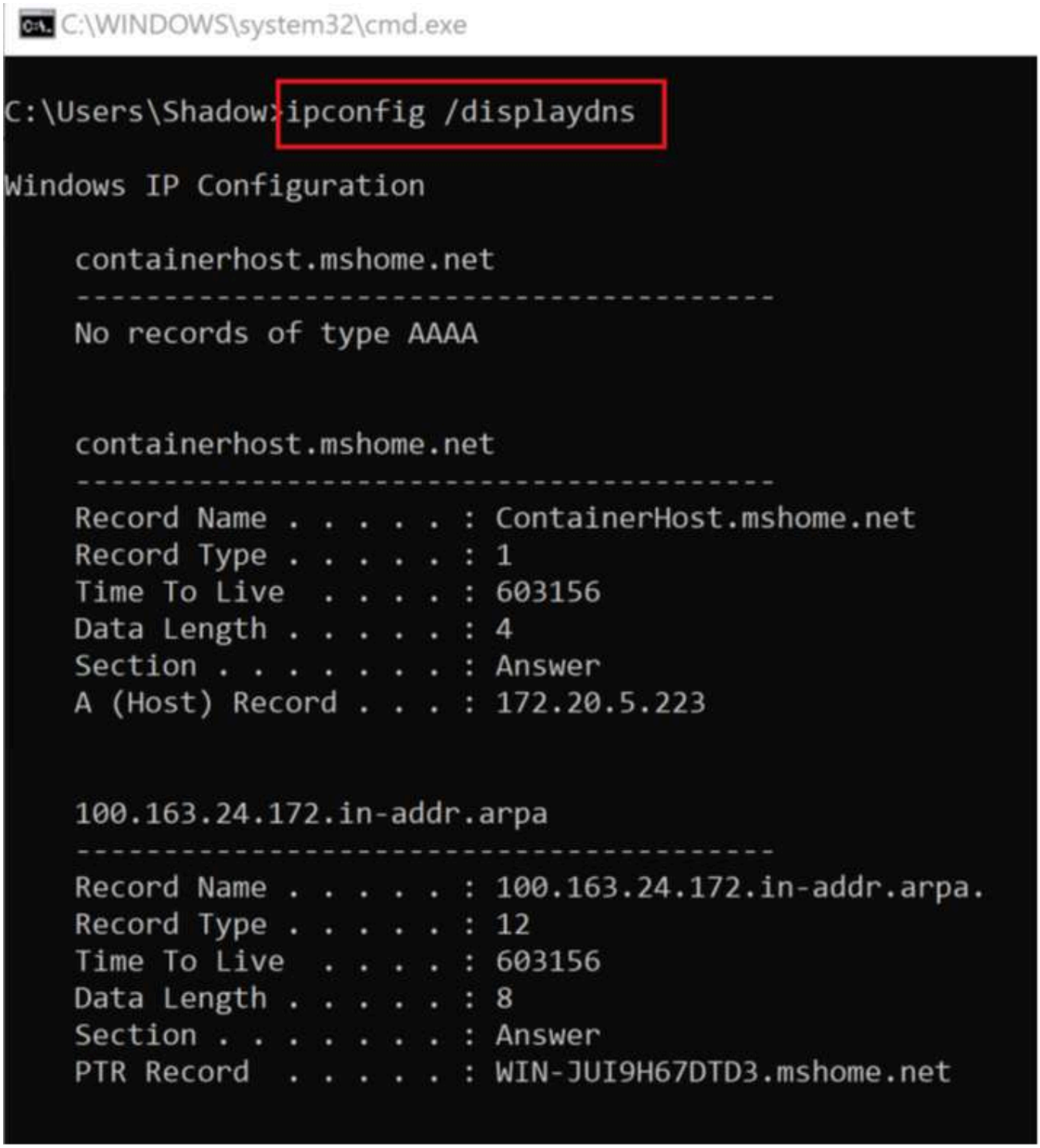
Releases a dynamic IP address.

Ipconfig /renew

Renews a dynamic IP address.

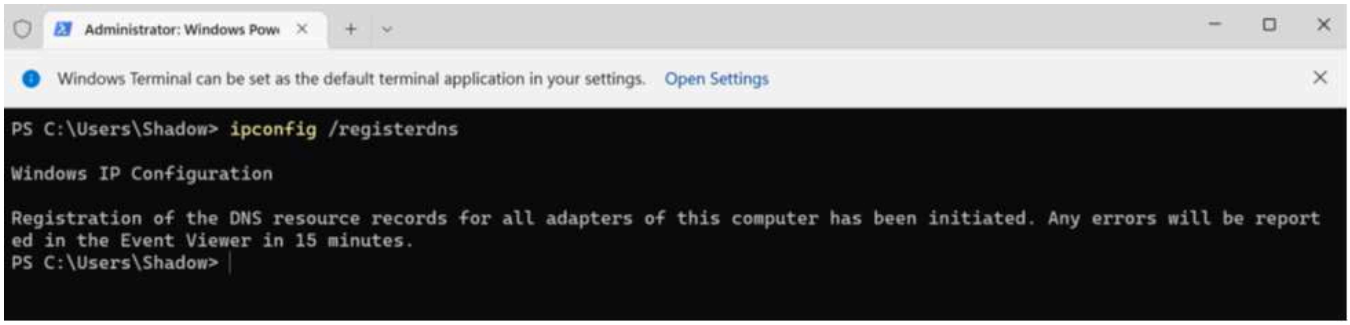
Ipconfig /displaydns

Displays the content of the DNS cache.

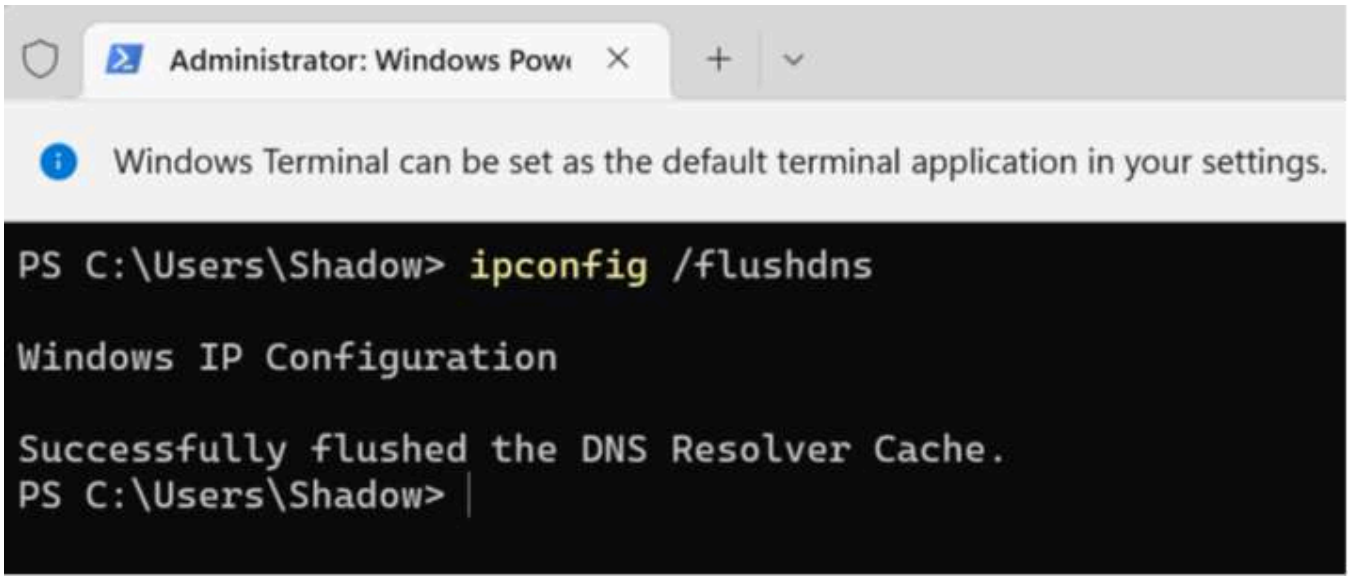


Ipconfig /registerdns

Causes the client to reregister its address with the DNS server.



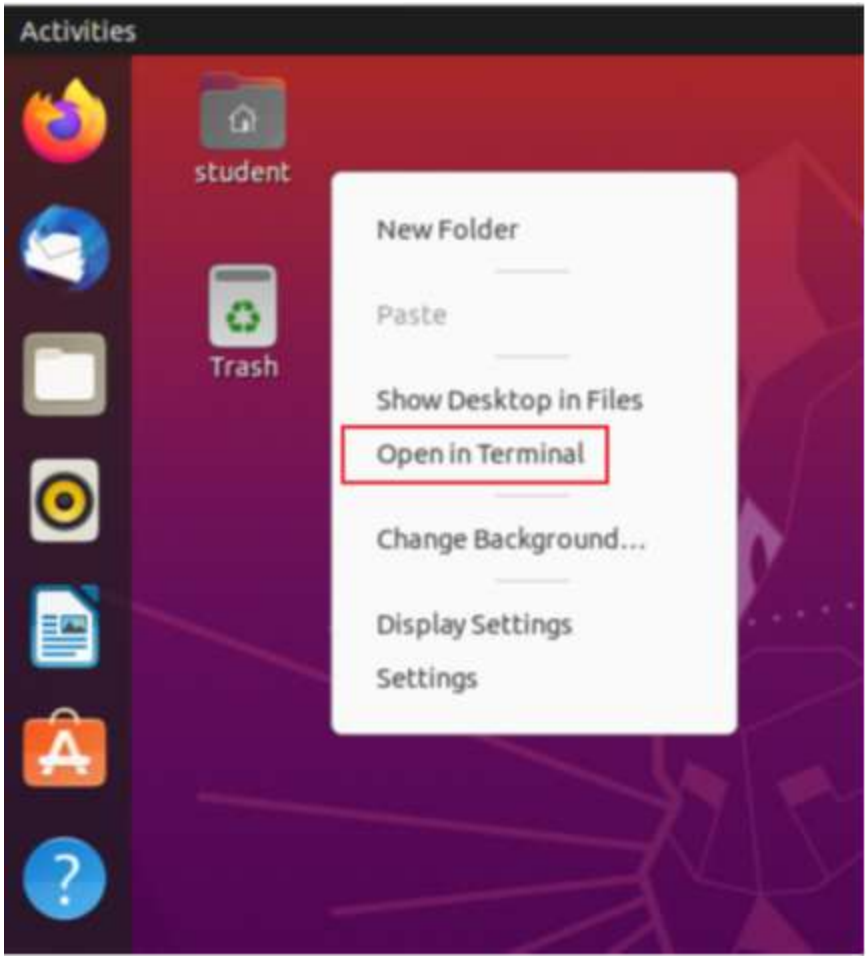
Ipconfig /flushdns
Clears the client DNS cache.



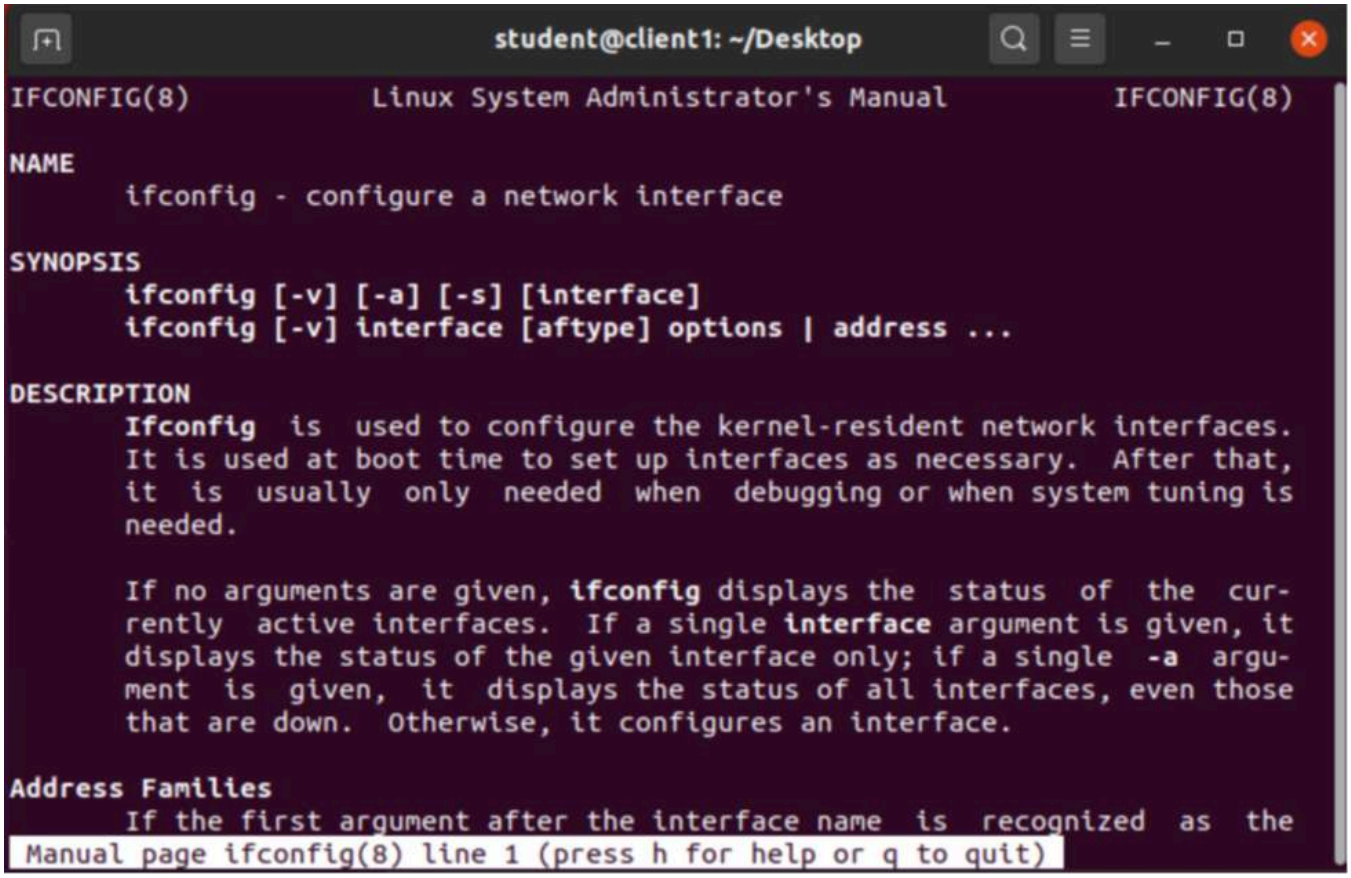
Note: If you need to find out more information about the NIC, beyond the IP address, subnet mask and default gateway, you need to use ipconfig /all. The most useful information from this command is the DNS server and DHCP server addresses, and the MAC address of the NIC.

Ifconfig

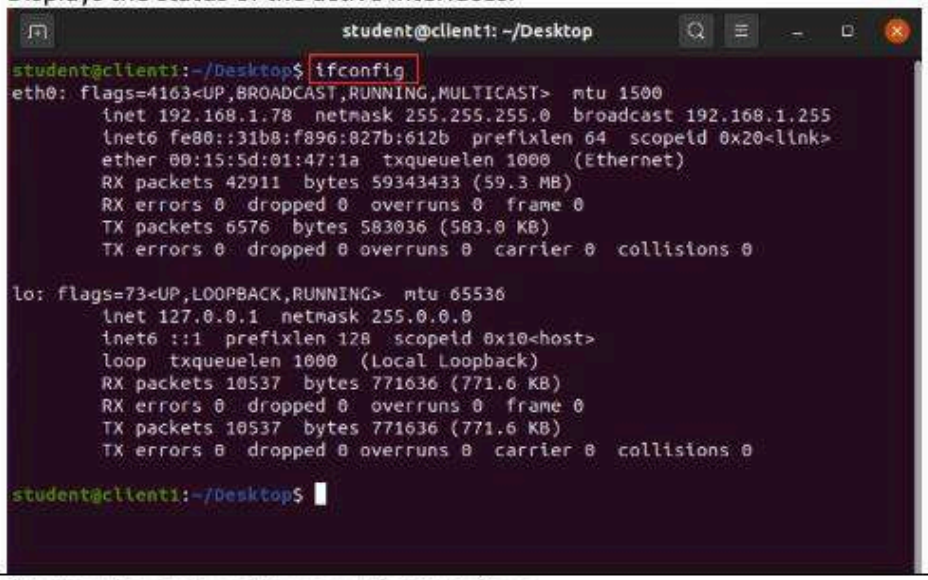
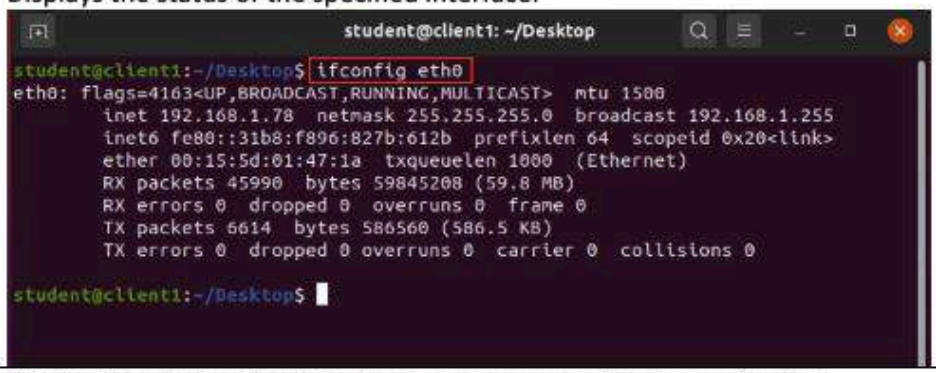
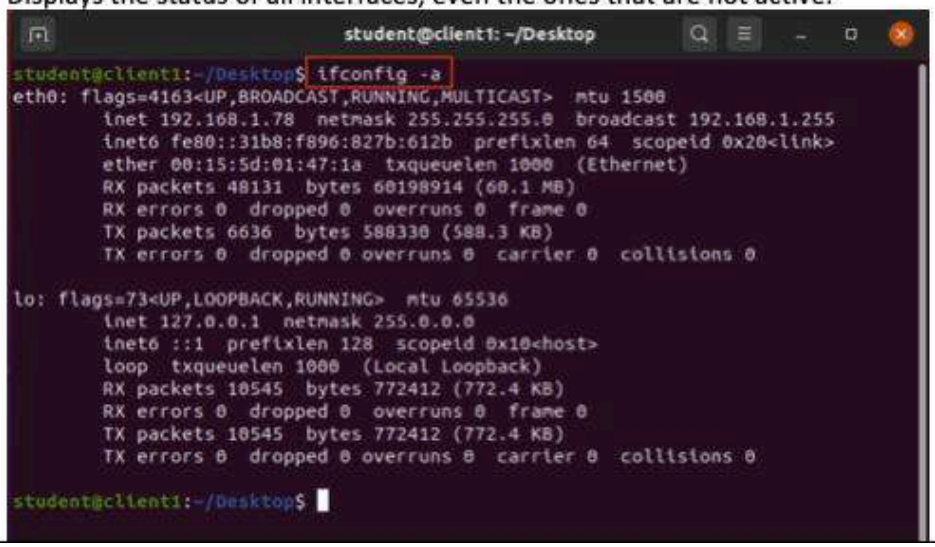
In Linux and in OS X, the equivalent command is ifconfig.
To open a terminal in OS X, from the Applications menu, choose Utilities and then terminal.
In Ubuntu Linux, you can right-click the **Desktop** and then click **Open in Terminal**.

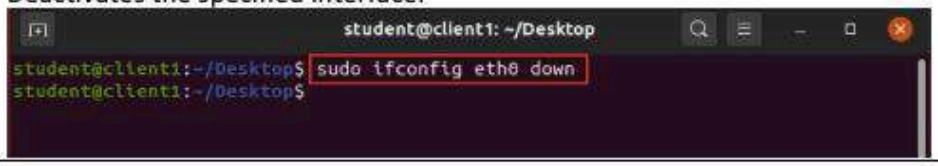
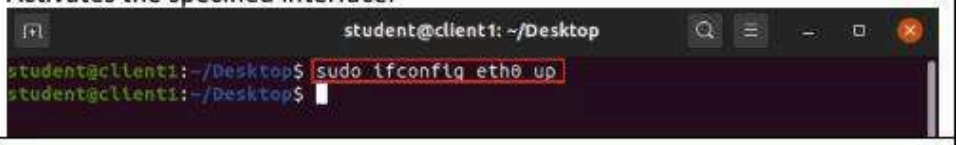
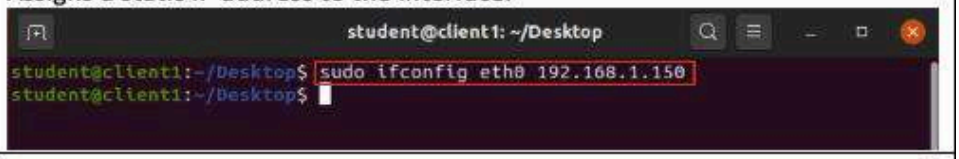
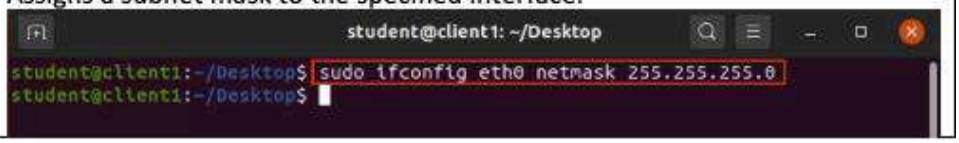


To get help on the ifconfig command, type **man ifconfig**.



The most common ifconfig switches you will us are:

Ifconfig	Displays the status of the active interfaces. 
Ifconfig *int*	Displays the status of the specified interface. 
Ifconfig -a	Displays the status of all interfaces, even the ones that are not active. 

sudo ifconfig *int* down	Deactivates the specified interface. 
Sudo ifconfig *int* up	Activates the specified interface. 
sudo ifconfig *int IPADDRESS*	Assigns a static IP address to the interface. 
sudo ifconfig *int* netmask *SUBNETMASK*	Assigns a subnet mask to the specified interface. 

Ping

The ping command is used to test round-trip connectivity. You can use ping with either a name or an IP address. I can ping akamai.com or I can ping that IP address and it's the same thing. Now ping uses the ICMP protocol. When you execute the ping command, the computer sends for ICMP echo request packets to the receiver. If there is round-trip connectivity and ICMP is not blocked by any firewalls between the sender and the receiver, the receiver will send back for ICMP echo reply packets and that's what we're seeing on the screen here. Keep in mind that ping only works if the sender can send the packet all the way to the receiver, and the receiver can send the reply all the way to the sender. If the packet arrives at the receiver, but the receiver cannot find the way to the sender, no reply will arrive. Now, if the sender doesn't get a reply from the receiver, you get different types of error messages. We can see request timed out or destination host unreachable. Let's check out one that's going to timeout. I'm going to ping 192.168.1.1, which is my default gateway or the address of the router. What I've done is I've actually told the router, do not respond to ping requests and so we're getting request timed out. Request timed out means the sender could find the receiver, but no reply was received. If the receivers on the local network usually just probably means that particular computer has ICMP blocked, which is the case here. If it's out on another network or remote network, it just means the reply didn't come back. Maybe it didn't come back in time, maybe it didn't

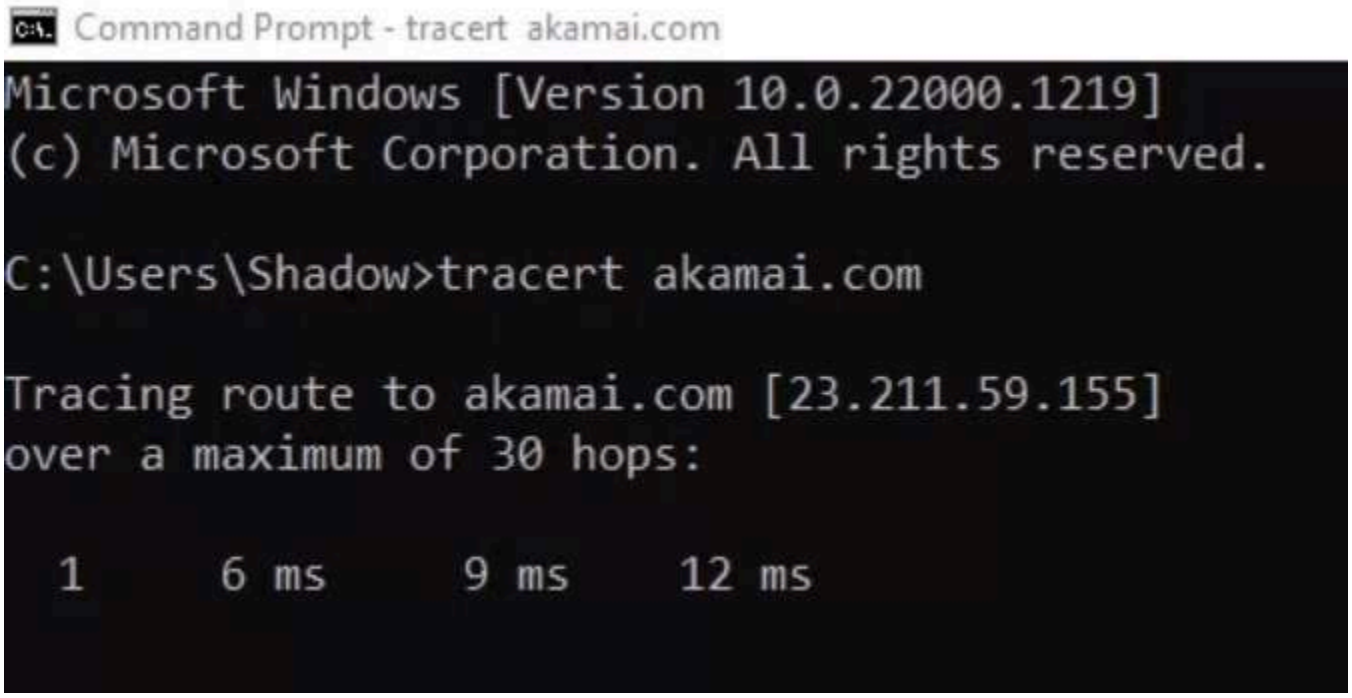
reply. It just tells us that the host knew what to do with it, but the reply didn't come back. Another message that you might see is destination host unreachable.

I'm going to try paying 192.1681.200. I know for a fact there is no device with that IP address, and so we're going to get destination host unreachable. In this particular case, it means that that particular device did not reply to an app request for the Mac address. If that was on a remote network, it would mean either that the client does not have a default gateway, or that the default gateway did not reply to the app request for a Mac address. Now there is one other situation where you might get a destination host unreachable message. When a packet is sent out on a network, it's configured with a time to live. We're actually just going to do a real quick trace route, we'll do akamai.com. You can see that second line over a maximum of 30 hops. So each time it crosses a router, we call that a hop. The packets are configured with a time to live. Here, the maximum will be 30. Then each time it crosses a router, the router deducts one from the time to leave. If that packet gets to a time to leave of zero, that router sends back a destination host unreachable and it drops the packet. The reason for this system is to prevent packets that have been sent to destinations that we can't get to or don't exist from being passed endlessly to routers across the network or the internet, so it doesn't just float around the Internet forever. You might see destination host unreachable. What it means is I could get to my default gateway, pass it to a bunch of other routers are gateways, but at some point, we're not going to get there. So it's unreachable because it's too many hops or there's just not a good route to that particular host. That's it for our discussion on ping. Ping uses the ICMP protocol. It sends out echo request and it gets echo replies. We could get replies, meaning we have round trip connectivity. We could get request timed out, meaning the remote host didn't respond either because ICMP is blocked or it just took too long or something went wrong with the communication. Or we could get destination host unreachable, which means that something didn't reply with its Mac address. Whether it's the default gateway or the other client, or maybe even that we don't have a default gateway at all.

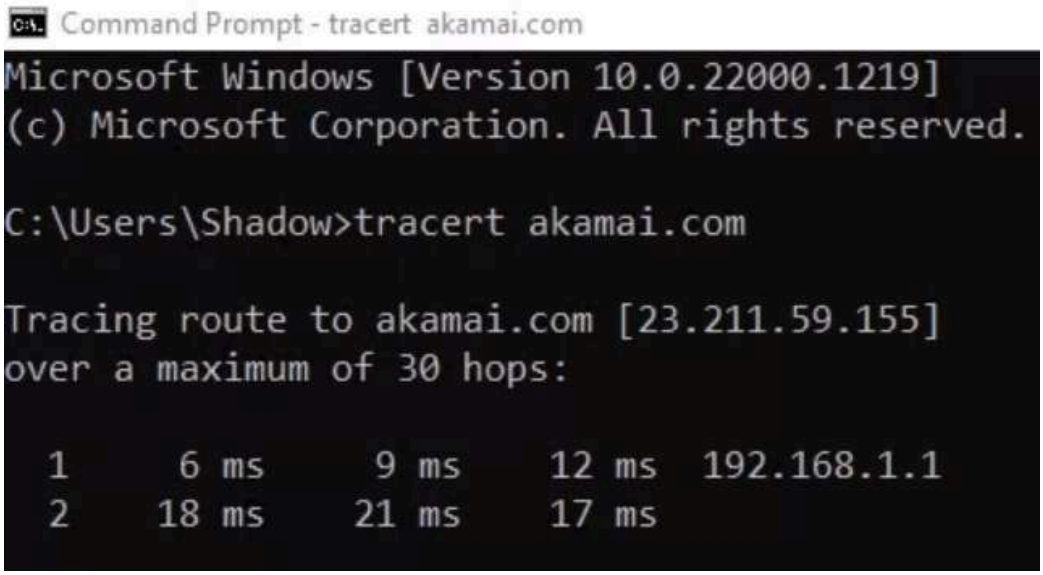
Tracert/Traceroute/Pathping

If the receiver isn't on the local network, it's sometimes useful to trace the path that the packet takes through the network. On a Windows operating system, the traceroute command traces the route the packet takes through the network to get the receiver. In an Windows operating system, it's actually spelled trace R-T but you just say traceroute. You can use traceroute with either a name or an IP address. We're going to do a traceroute to akamai.com. Traceroute uses a series of ICMP echo requests to trace the route through the network. You can see it comes up over a maximum of 30 hops. Each hop is a different router. The first

packet sent out has a time to live of one. It arrives at the first router, the router deducts one from the time to live and the packet expires so that first router sends back a destination host unreachable message with its IP address and name if it has one. That generates that first line of output.



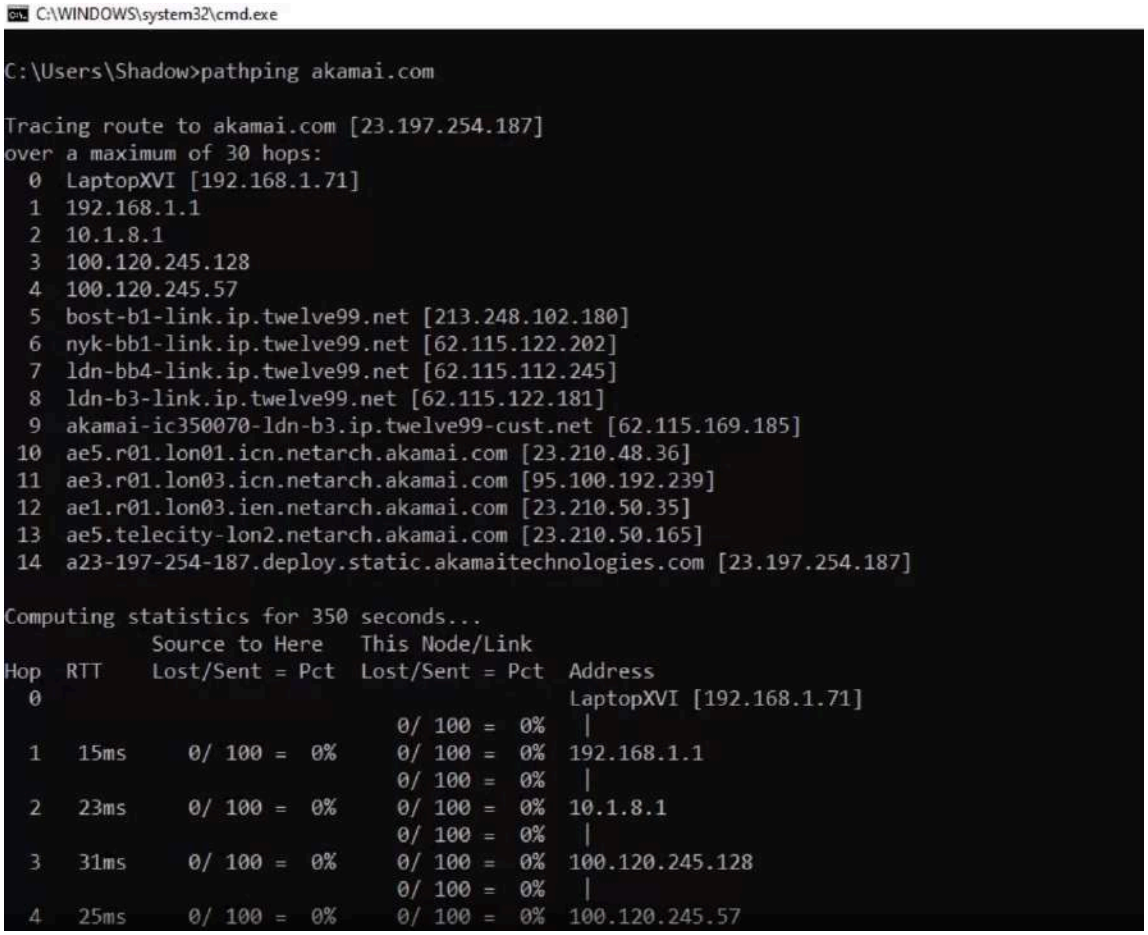
Then the computer sends out a second ping with a time to live of two. It passes through the first router and expires at the second router which sends back its name and IP address and that generates a second line. Traceroute keeps increasing the time to live until the packet arrives at the destination. You can speed up this a little faster, you do not have to make it take so long, it's just going to be very slow on my computer so I'm just going to be quiet and then once it's done, I'll start talking again.



Our traceroute is complete. Traceroute uses the ICMP protocol because it's using a series of pings and pings use ICMP. If you take a look, you can see that in hop 9, we see Asterisk instead of the time and we get a request timed out. That's a router that has ICMP disabled and does not respond to a ping. Because ICMP has been used in some network attacks, some routers have it disabled for security. One of the things that's really interesting that I found in testing out this video before I recorded it, is for many years, I've always used Yahoo as a way to demonstrate ping in classes because Yahoo has always responded for ping. I never used Microsoft or Google because they didn't respond.

On a Linux or an OS 10 machine, the command is traceroute, but it's spelled like this.On Windows, it's tracert but you say traceroute for both of them

The other command we want to take a look at is path ping. Here you can see I did a path ping to akamai.com. Path ping works similar to traceroute but as you can see, it supplies much more extensive information on the path through the network. In the beginning, it executes a simple traceroute but then it measures statistics like packet loss and the timing of each hop and you can see over here, there were 14 hops when I ran the path ping and it computed statistics for 350 seconds which is almost six minutes. The amount of time it takes to run a path ping depends on how many hops there are and the speed of the network.



This could take 10 minutes or more to run. It's not one that I worked with a lot because I usually don't have the patience to sit there for 10 minutes but if I really needed to know about packet loss and more about speed that kind of thing, you can see here the speed jumps up radically in these particular hops it's just interesting. Then I can get much more information. This can be useful but it does take quite some time to complete so I would really be having to troubleshoot some type of network slow down beyond the local network before I would try something like pathping. That's it for this video. In this video, we looked at traceroute, which traces the route a packet takes through a network by using a series of ICMP ping requests with varying time to live so that they expire at each successive router and send back the information and pathping which is like a traceroute but it provides network statistics like packet loss and timing.

TCP/IP Services Lab

Configure DHCP and DNS

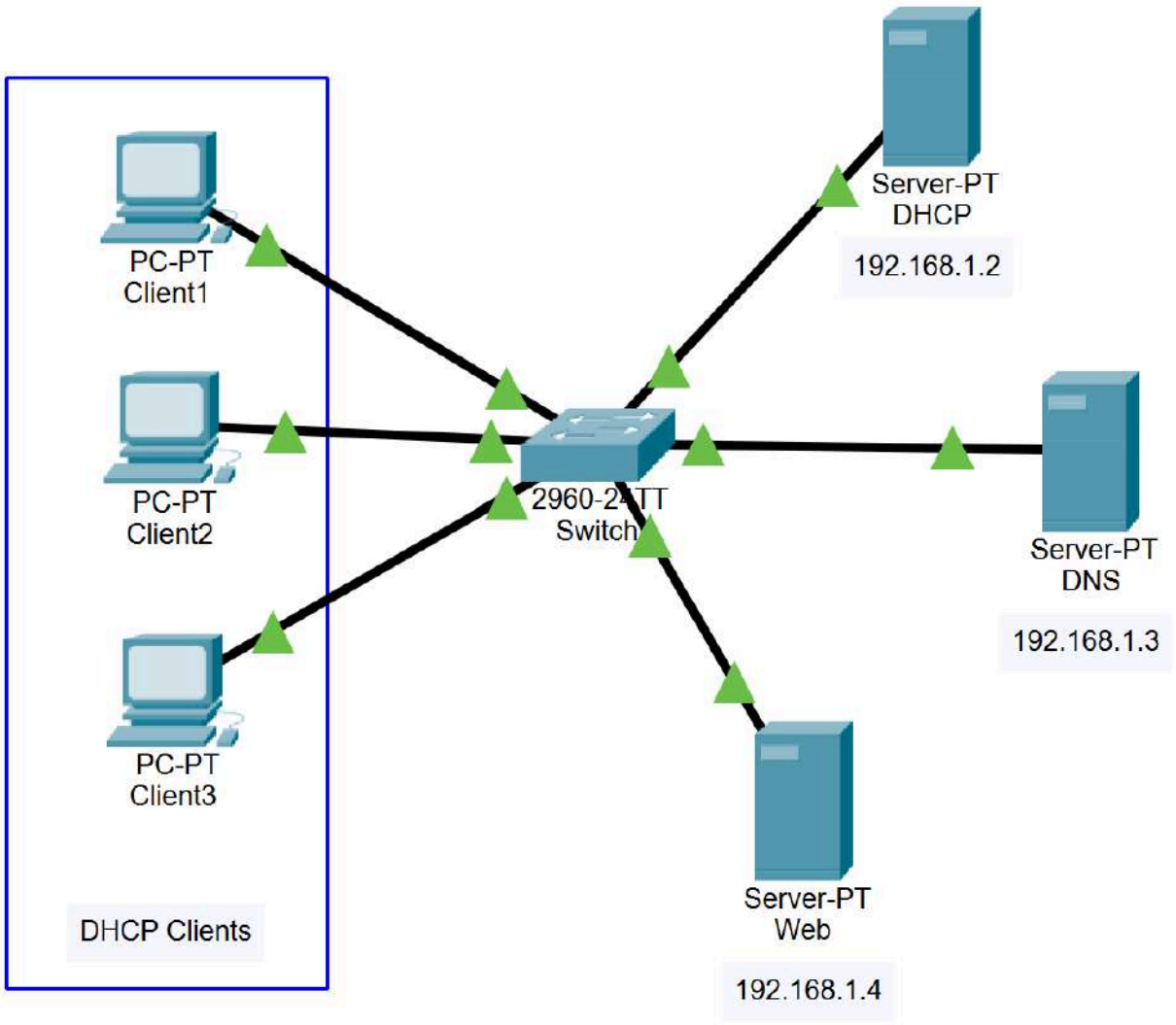
In this lab you will investigate APIPA addresses on clients set to obtain an IP address via DHCP in a network where DHCP is not present. While there is a server named DHCP in the sample file, it will not provide DHCP services until you configure it later in the lab.

TASK A

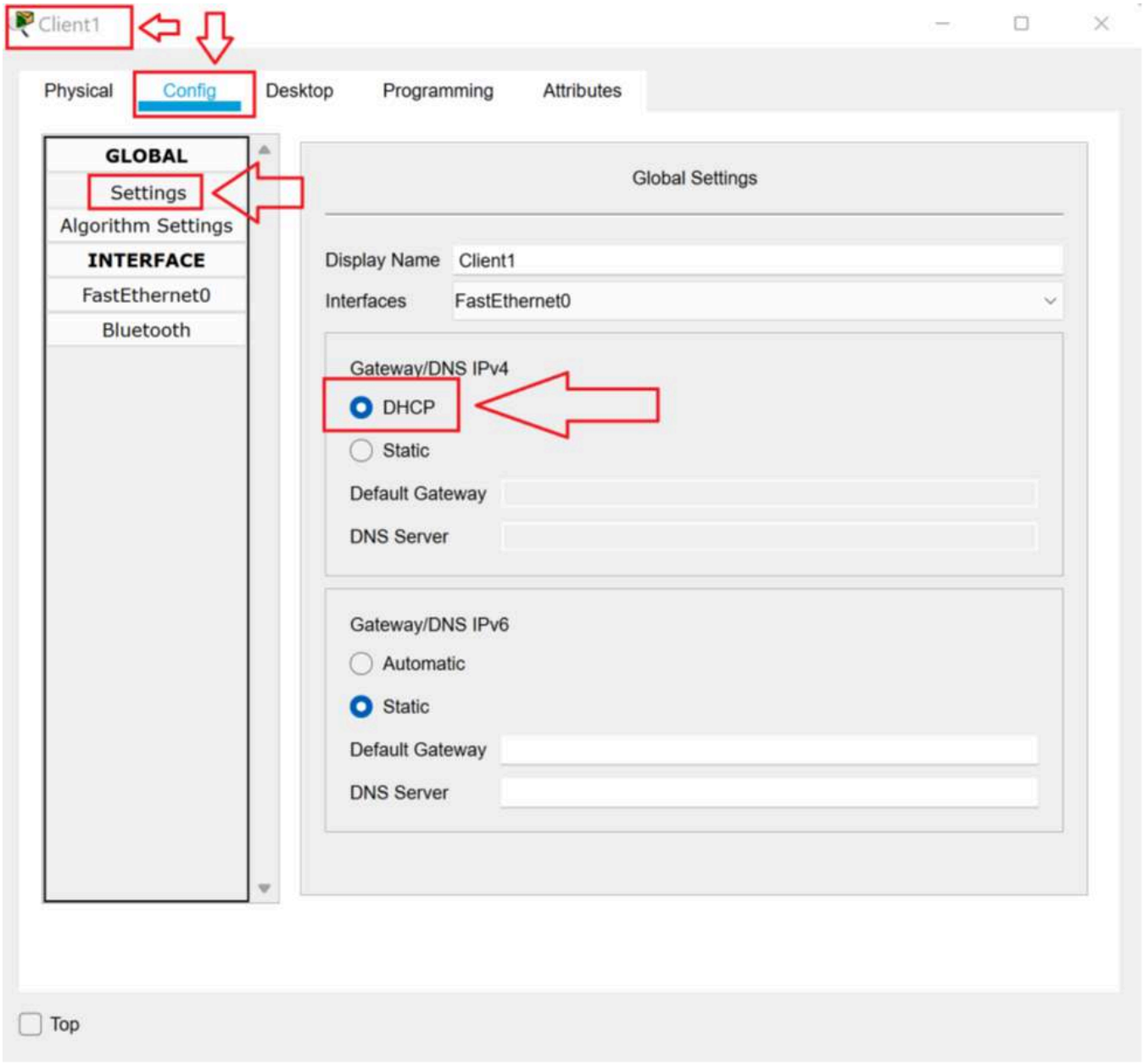
In this task, you will look at APIPA.

1. Download the **5.4.1 Lab File** and open it in **Packet Tracer**.

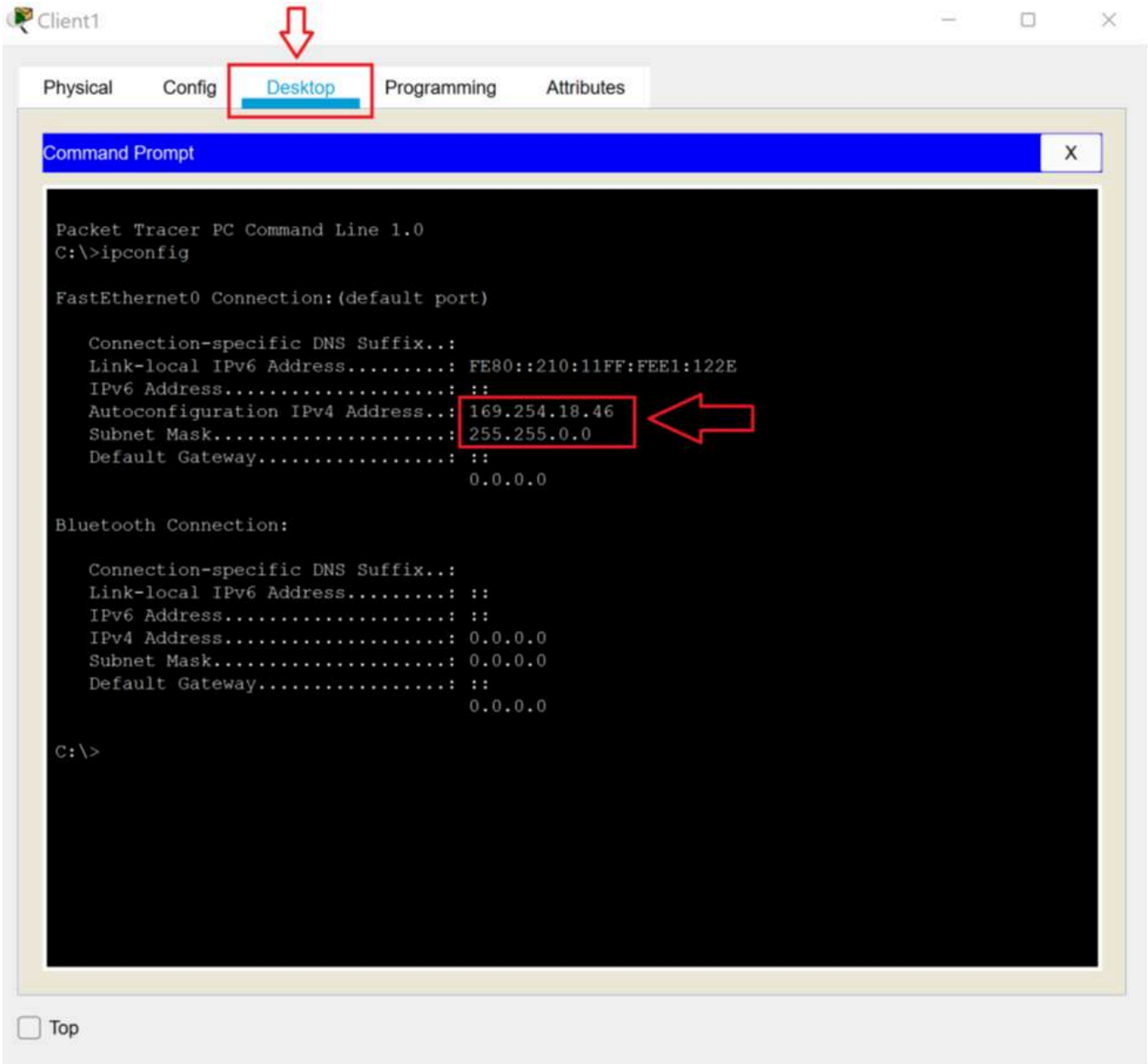
5.4.1 Lab File
PKT File



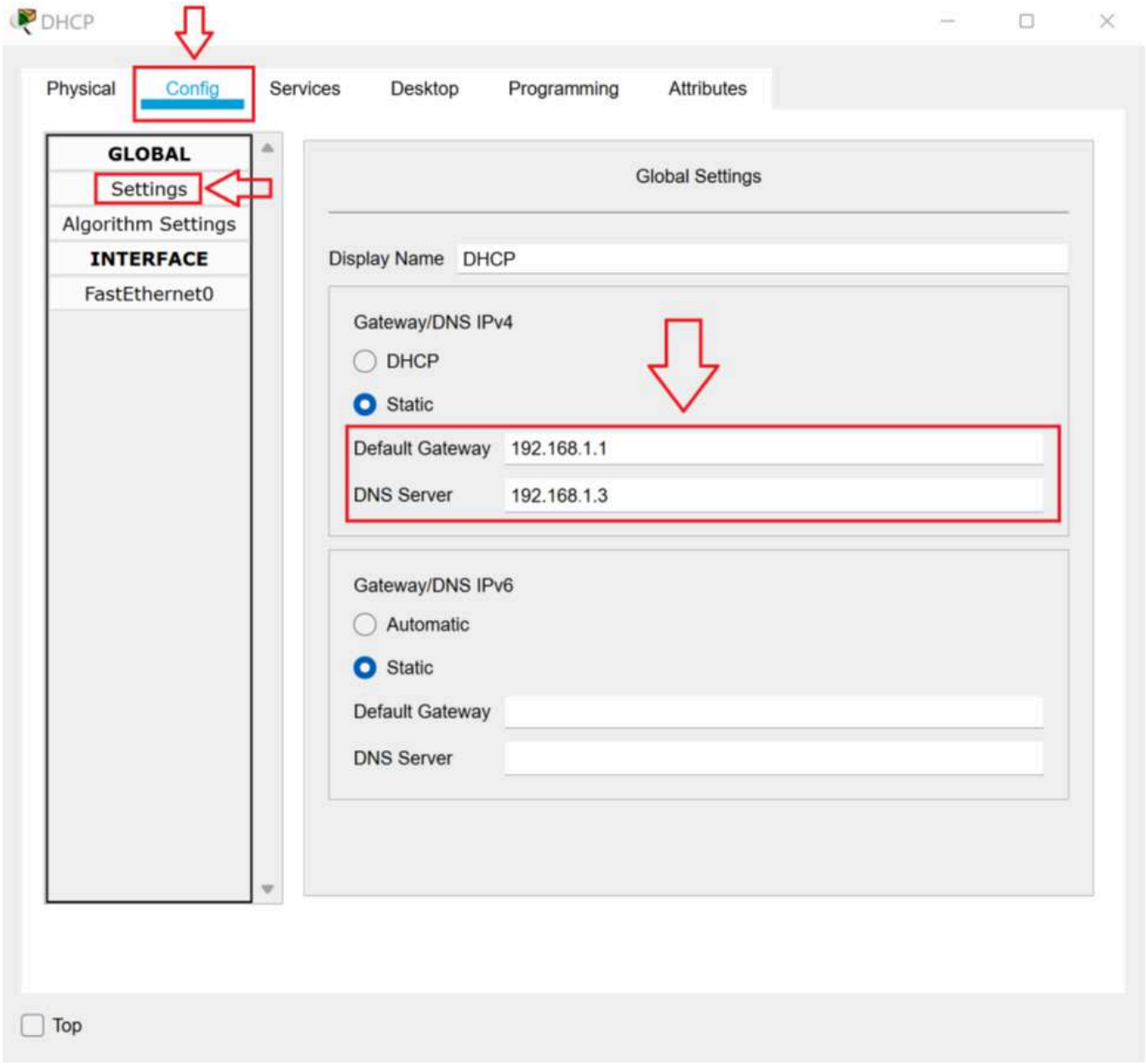
2. Click **Client1** to open the **Client1 Properties** dialog box. In the **Config** tab, under the **Global** menu on the left, click **Settings**. Observe that in the **Gateway/DNS IPv4** settings, the **DHCP** radio button has been selected.



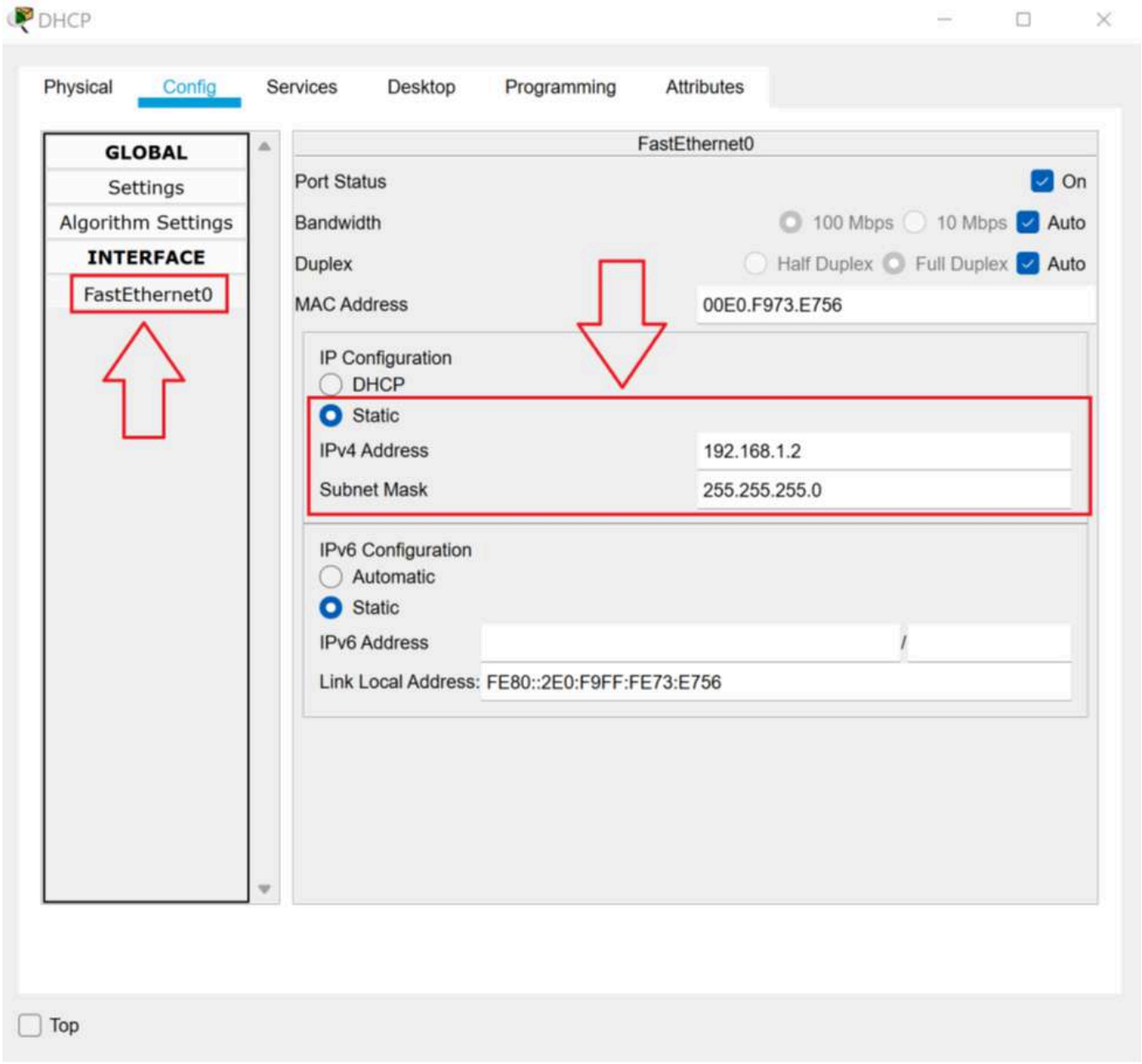
3. Select the **GUI** tab, then click the **Command Prompt** icon. In the **Command Prompt**, execute the **ipconfig** command. Notice the client has an APIPA address.



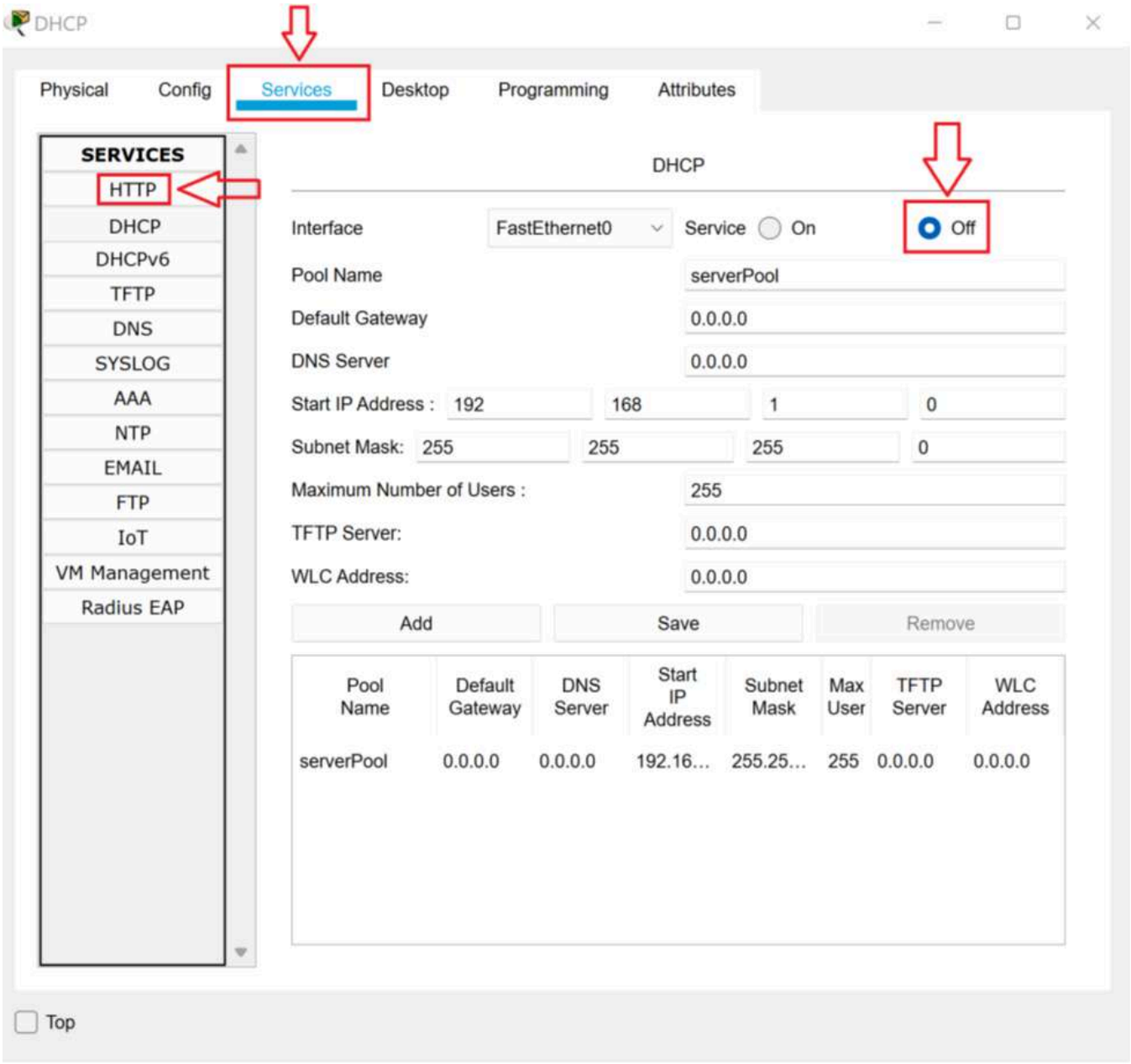
4. Close the **Client1 Properties** dialog box.
5. Click the **DHCP** server to open the **DHCP Properties** dialog box. On the **Config** tab, under the **Global** menu on the left, click **Settings**. Observe that the **Default Gateway** is set to **192.168.1.1** and the **DNS Server** is set to **192.168.1.3**.



6. Under the **Interface** menu on the left, click **FastEthernet0**. Observe that the **IP Configuration** is set to the **Static** radio button. The server is configured to use an **IPv4 Address** of **192.168.1.2** with a **Subnet Mask** of **255.255.255.0**.

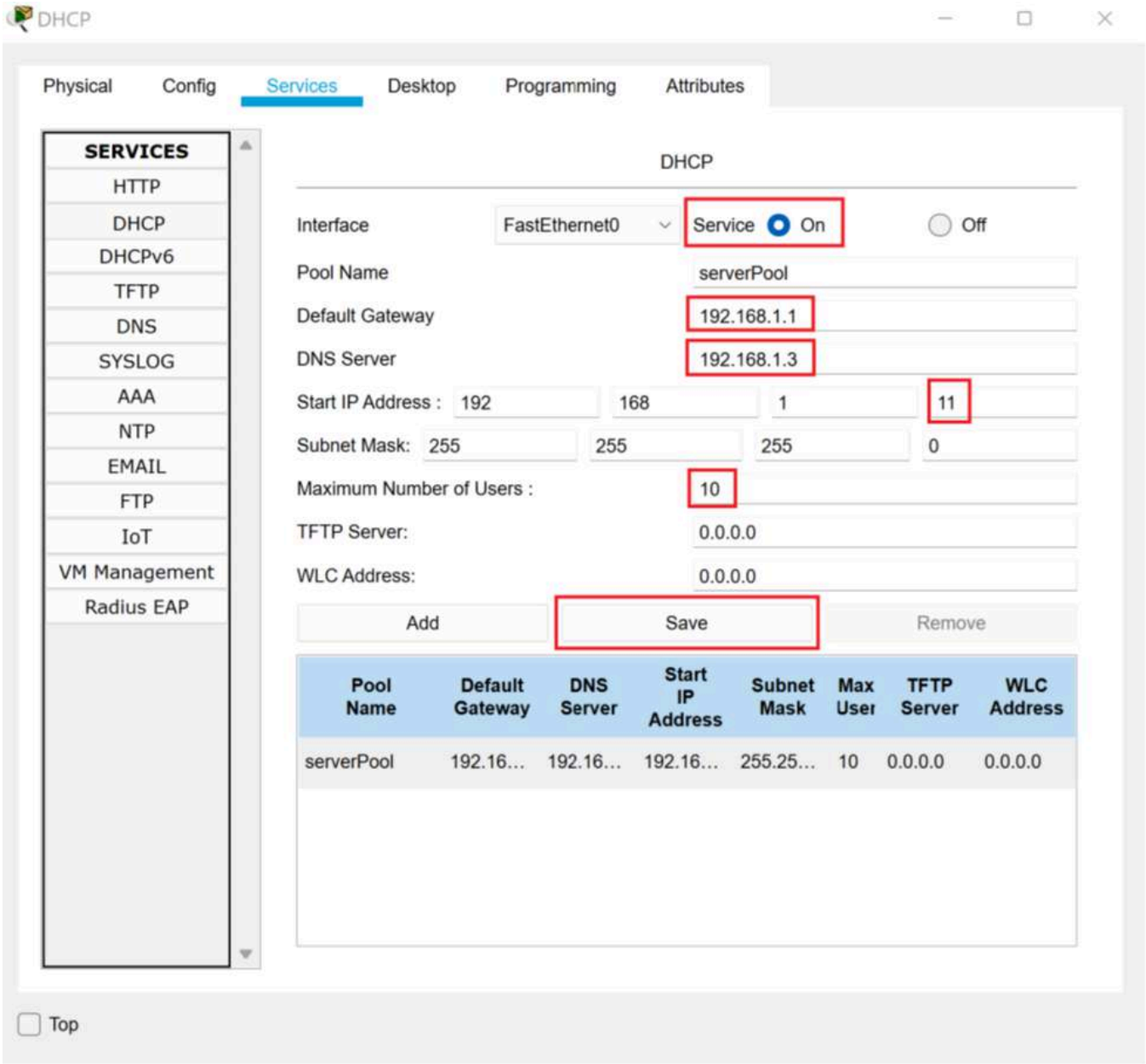


7. Click the **Services** tab. In the menu on the left, click **DHCP**. Observe that the **Service** radio button is set to **Off**.

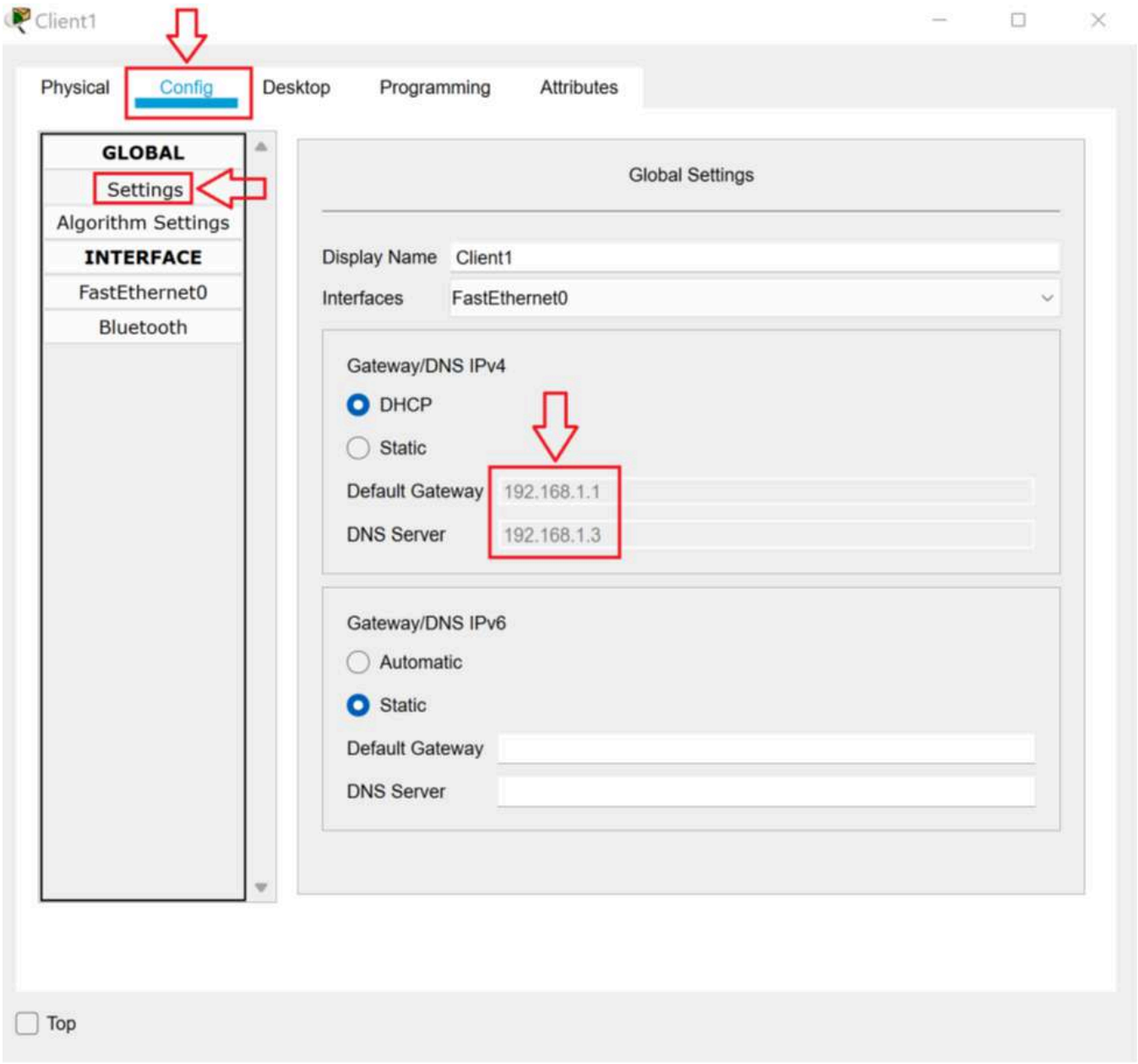


TASK B

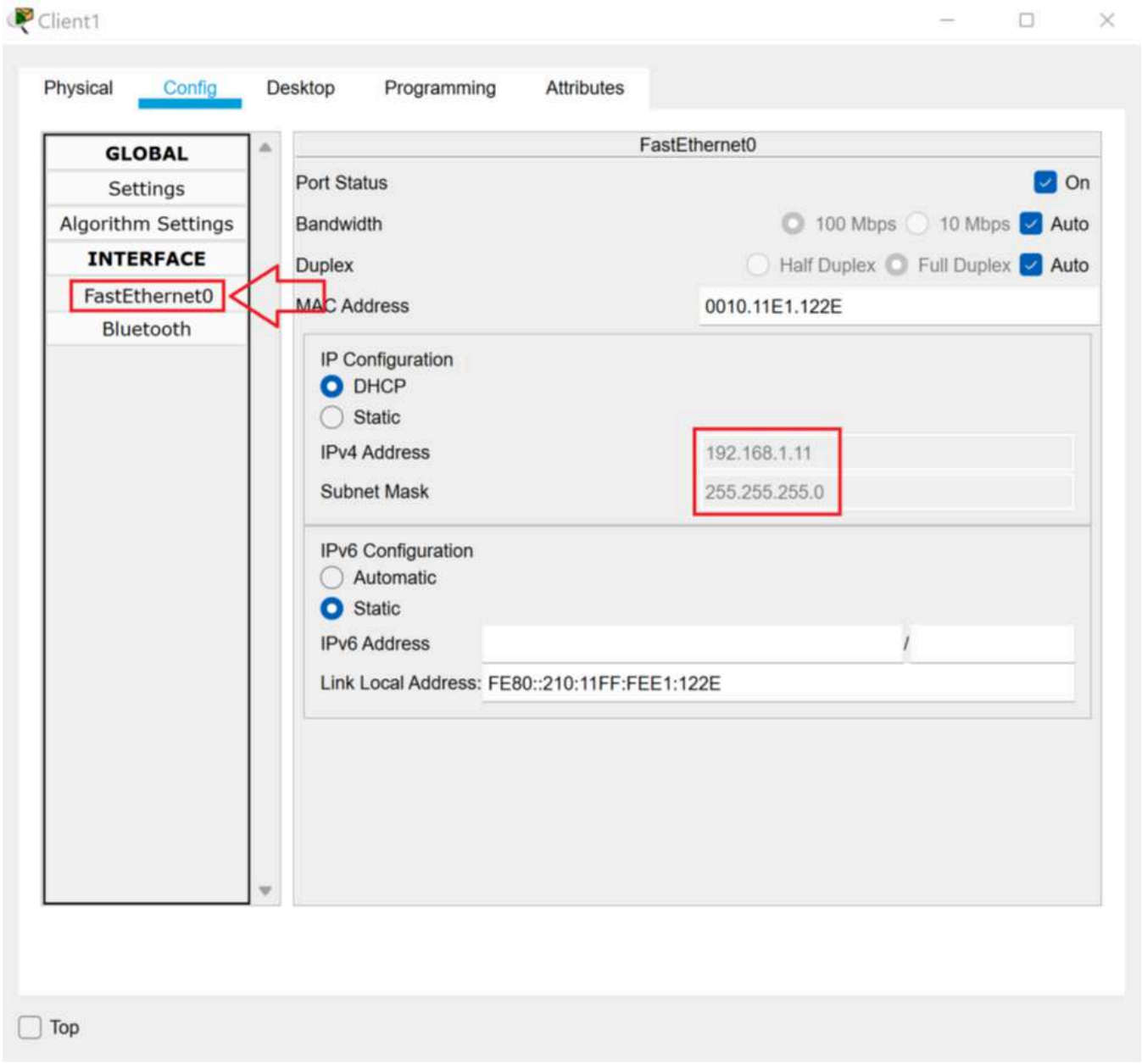
To resolve the issue in the network, you will configure DHCP to issue IP addresses to the clients.
1. If the **DHCP Properties** dialog box is not open from the previous task, click the **DHCP** server to open the **DHCP Properties** dialog box. On the **Services** tab, click **DHCP**.
2. In the **Default Gateway** text box, enter **192.168.1.1**. In the **DNS Server** text box, enter **192.168.1.3**. In the fourth octet of the **Start IP Address** enter **11** so that the entire address is **192.168.1.11**. In the **Maximum Number of Users** text box, enter **10**. In the **Service** section, select the **On** radio button. Click the **Save** button.



3. Click **Client1** to open the **Client1 Properties** dialog box. On the **Config** tab, in the **Global Settings**, notice that in the **Gateway/DNS IPv4** section, the DHCP server has provided a **Default Gateway** of **192.168.1.1** and a **DNS Server** of **192.168.1.3**. The settings are grayed out because they cannot be changed at the client.



4. Select the **FastEthernet0** tab. Notice that in the **IP Configuration** section, the DHCP server has provided an IP address and subnet mask from the pool.



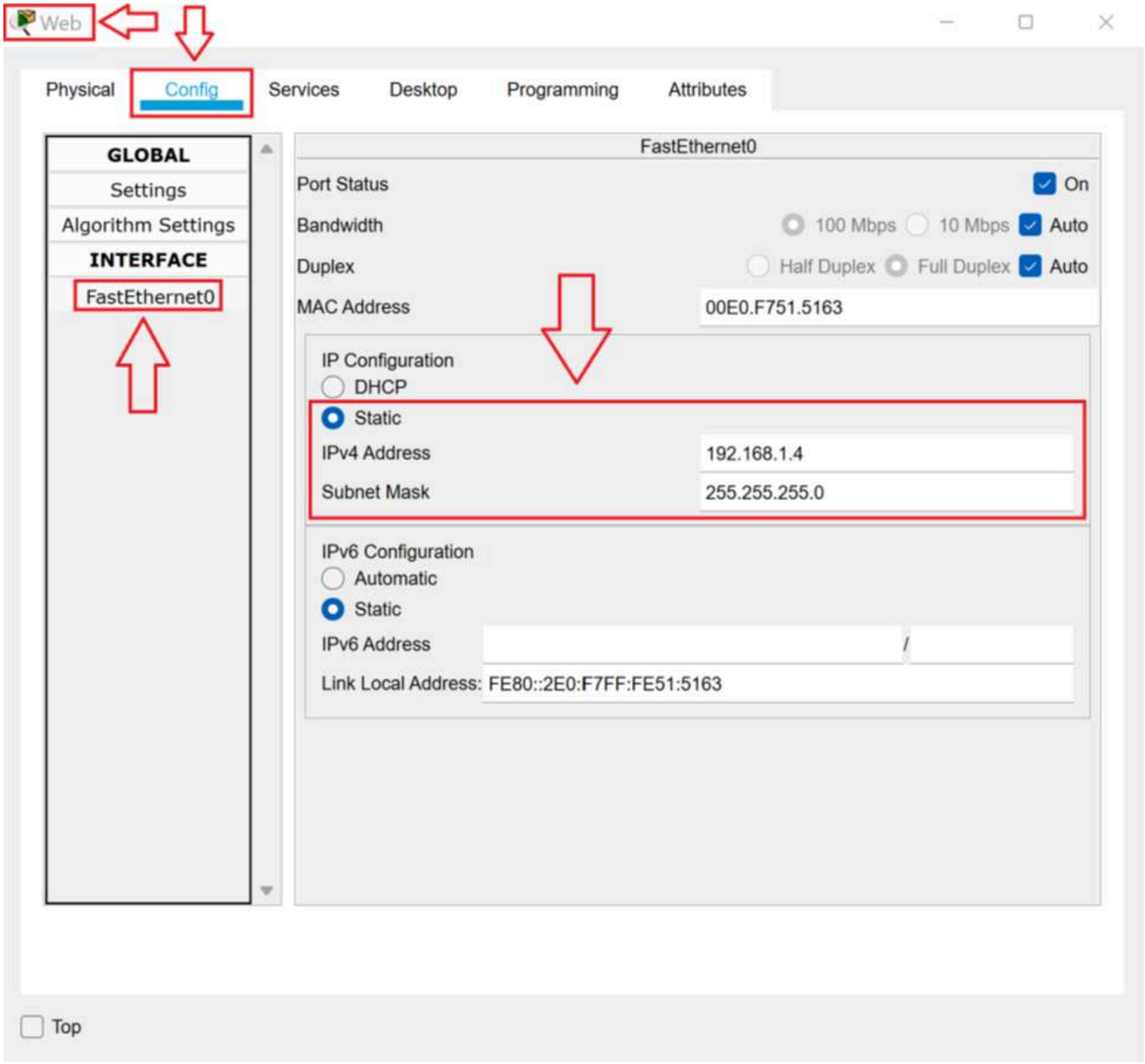
Note: If the client does not show an address from the pool, click the **Static** radio button then click back on the **DHCP** radio button to reset the card.

5. Close the **DHCP Properties** dialog box.

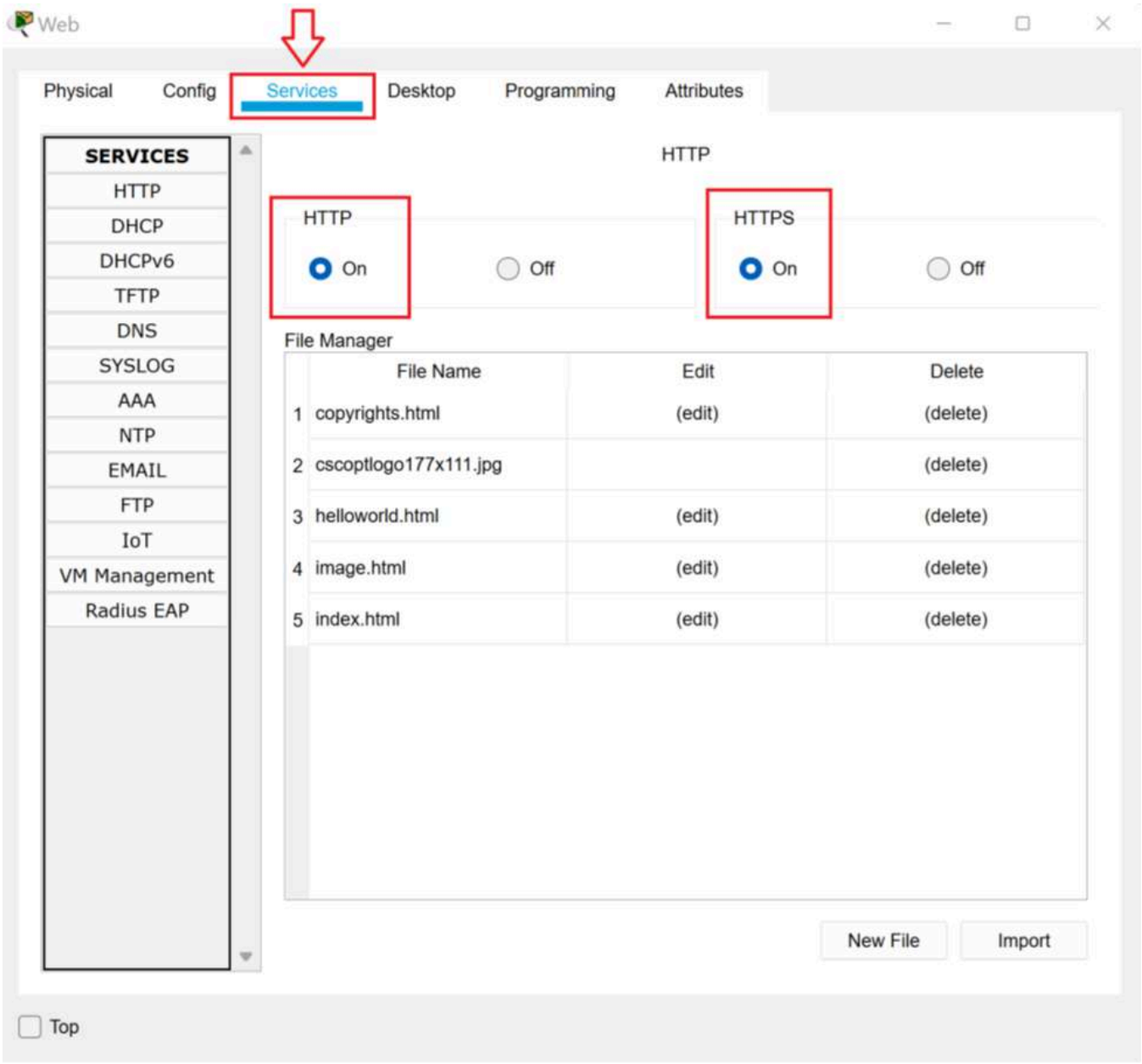
TASK C

Now that you have configured DHCP, the DHCP clients have valid IP addresses and can access all the devices in the network by IP address. Now, to make it easier for the clients, you will configure DNS so they can reach the company Intranet using a friendly name.

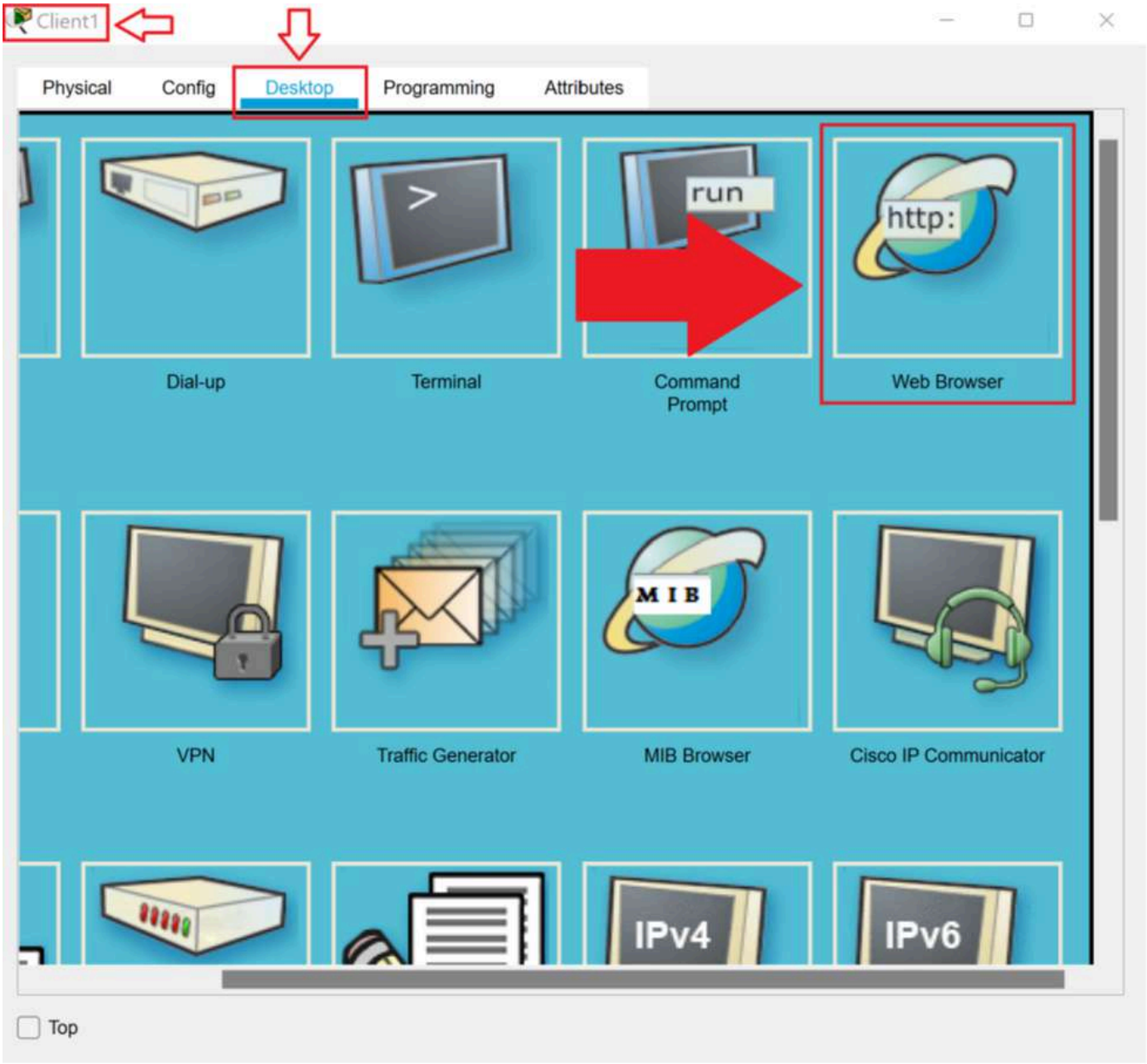
1. Click the **Web** server to open the **Web Properties** dialog box. On the **Config** tab, click **FastEthernet0**. Observe that the server has been assigned a **Static IP Address** of **192.168.1.4** with a **Subnet Mask** of **255.255.255.0**.



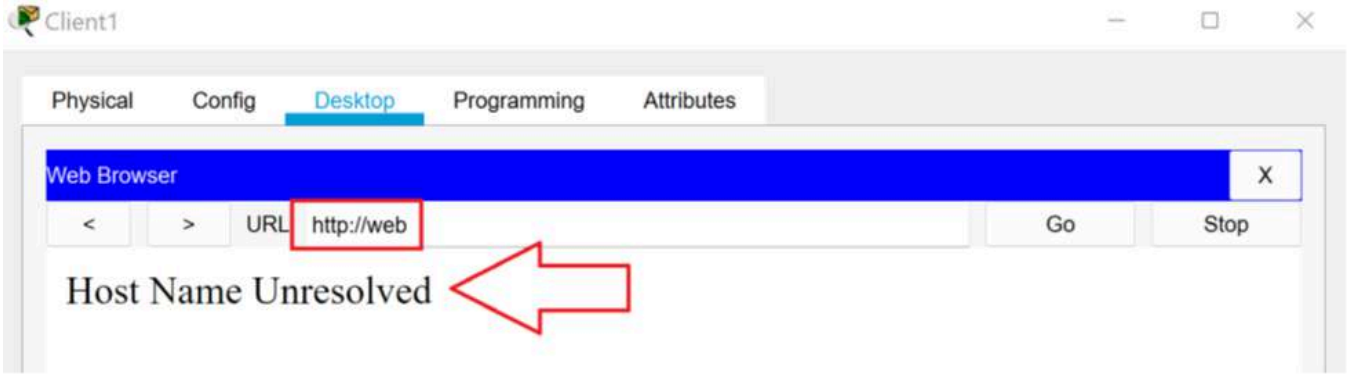
2. On the **Services** tab, select **HTTP**. Notice that both **HTTP** and **HTTPS** are set to **On**.



3. Close the **Web Properties** dialog box.
4. Click **Client1** to open the **Client1 Properties** dialog box. On the **Desktop** tab, scroll to the right and then click the **Web Browser** icon.

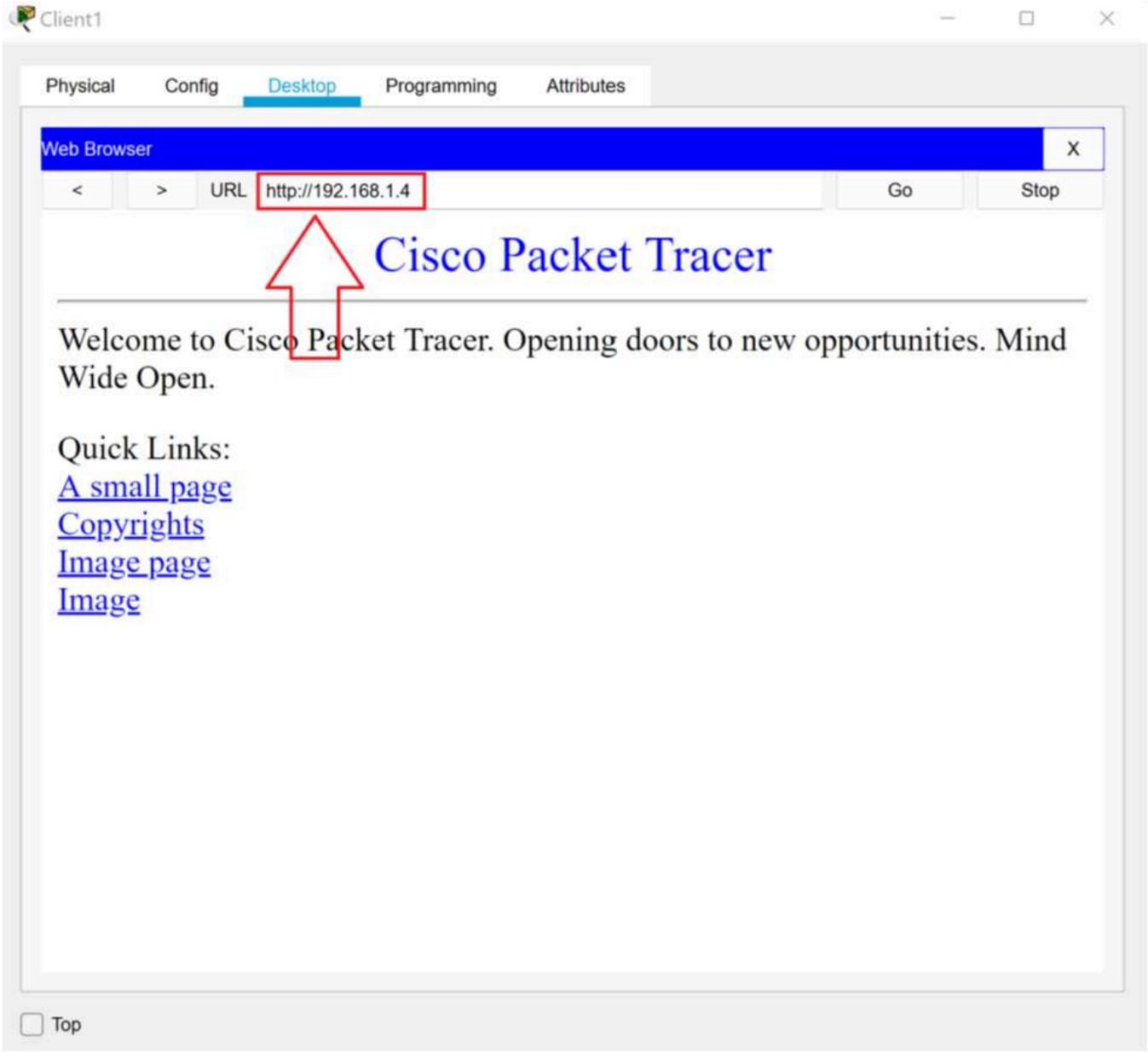


5. In the **URL** text box, type **http://web** and press **Enter**. Observe that this does not retrieve a web page.



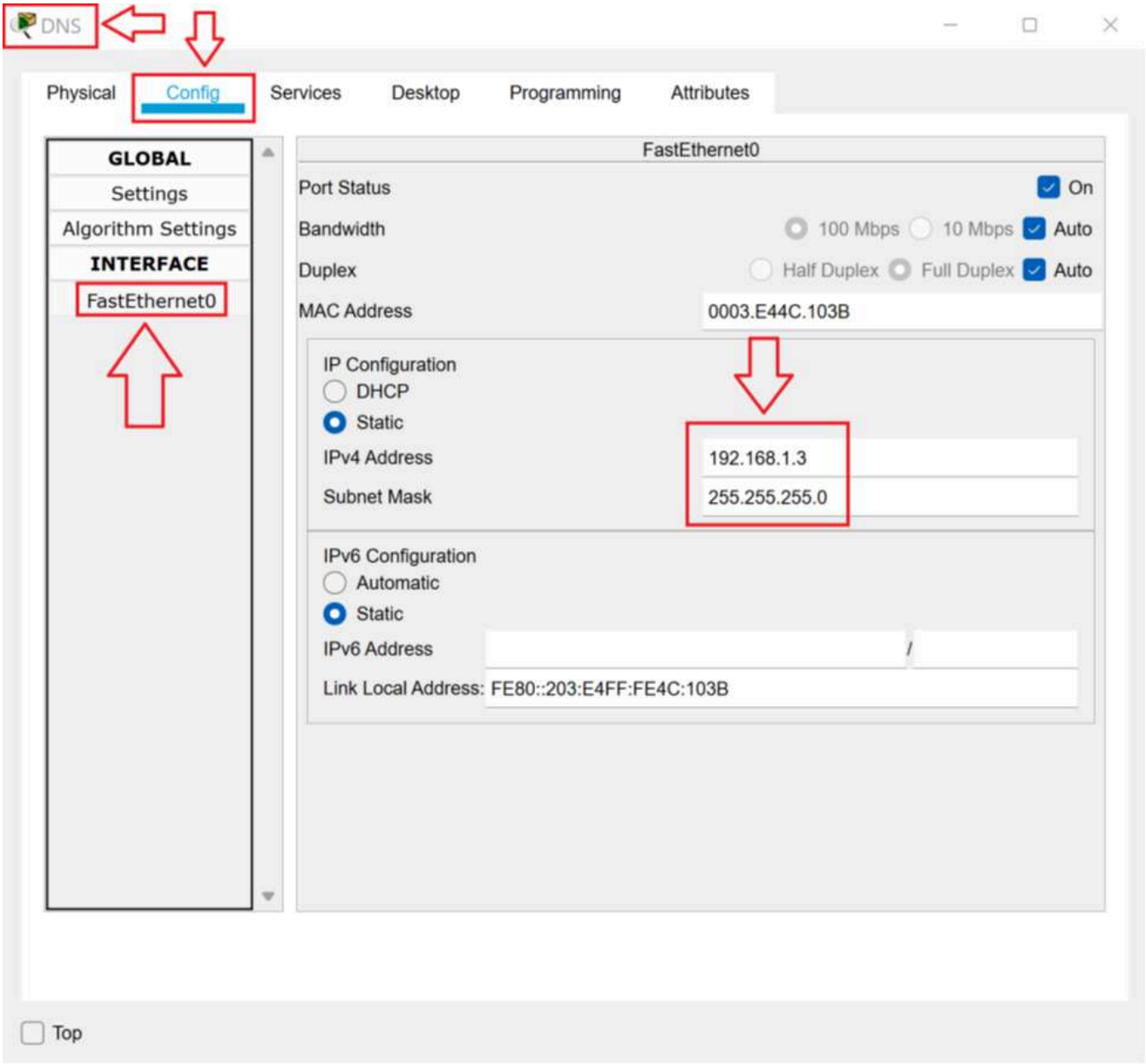
6. In the **URL** text box, type in **http://www.company.com** and press **Enter**. Notice this request also fails.

7. In the **URL** text box, type in **http://192.168.1.4**. Notice that the web page loads in the browser.

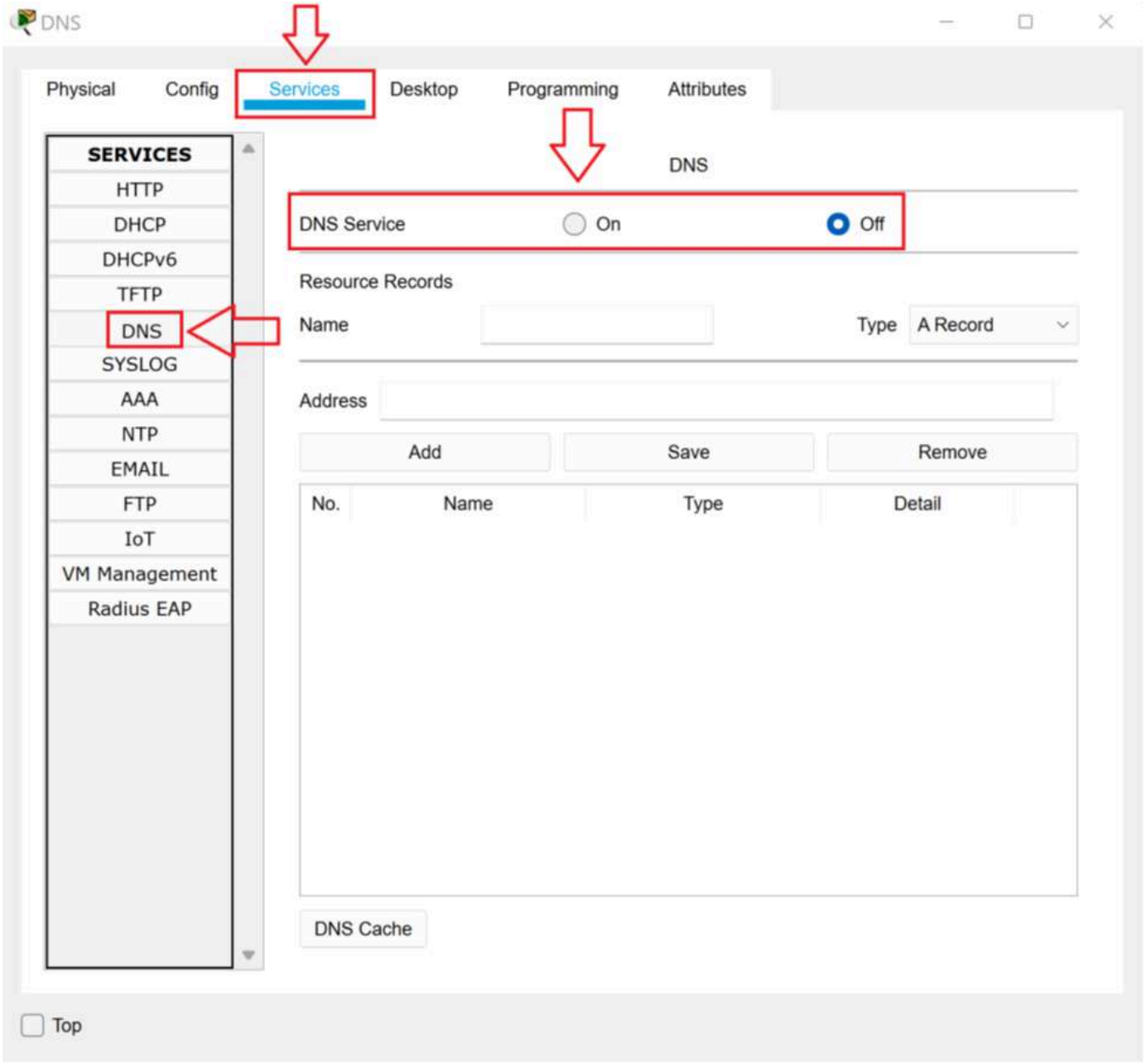


8. Close the **Client1 Properties** dialog box.

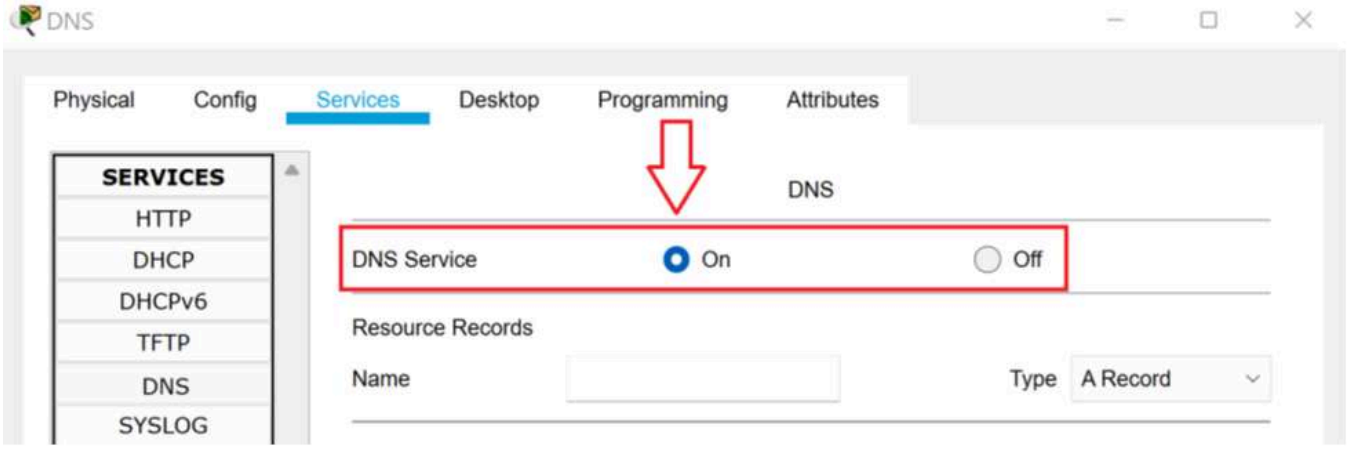
9. Click the **DNS** server to open the **DNS Properties** dialog box. On the **Config** tab, select **FastEthernet0**. Observe that the server has been assigned a **Static IP Address** of **192.168.1.3** with a **Subnet Mask** of **255.255.255.0**.



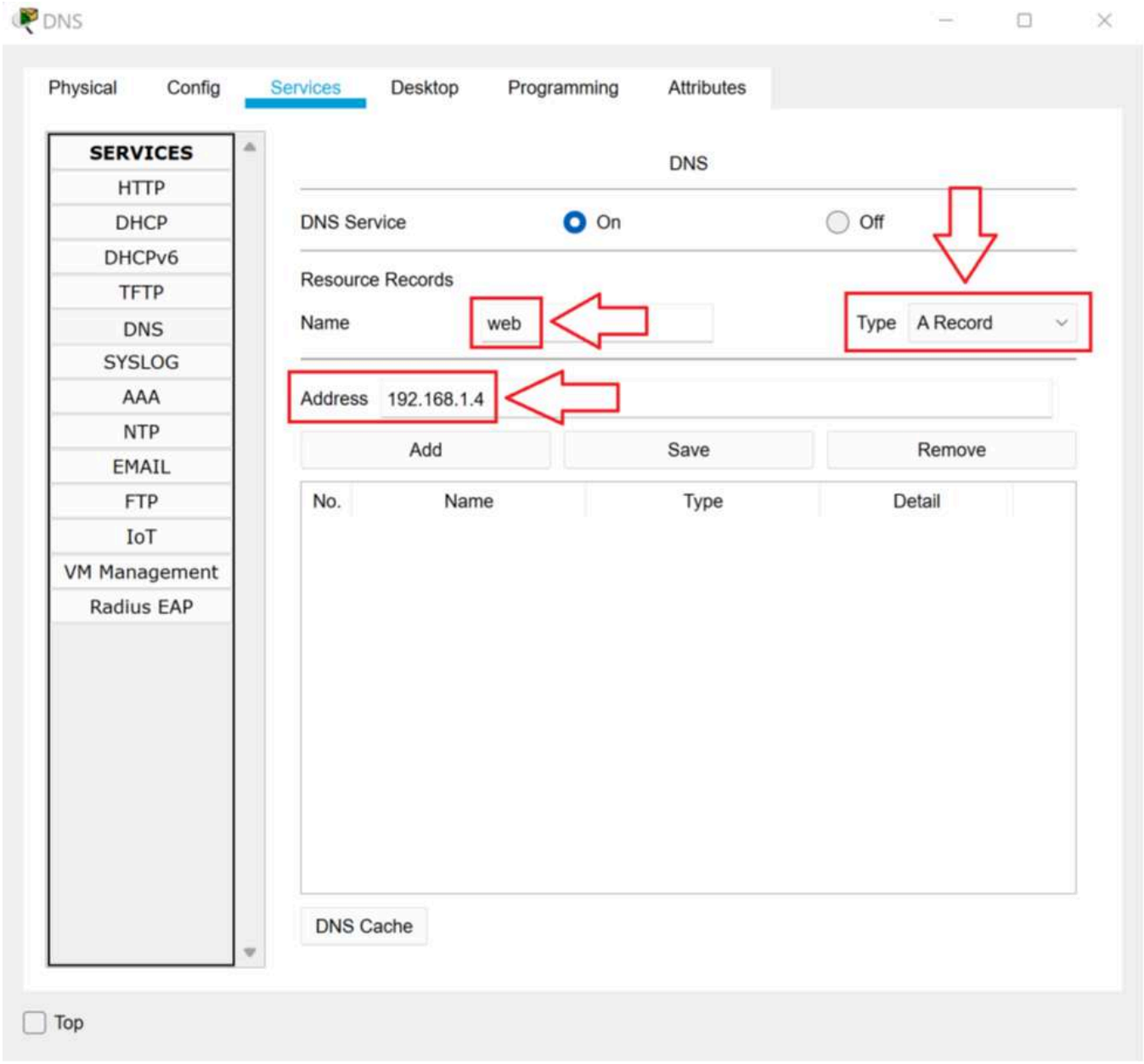
10. On the **Services** tab, select **DNS**. Notice the **DNS Service** is set to **Off**.



11. In the **DNS Service** section, select the **On** radio button.

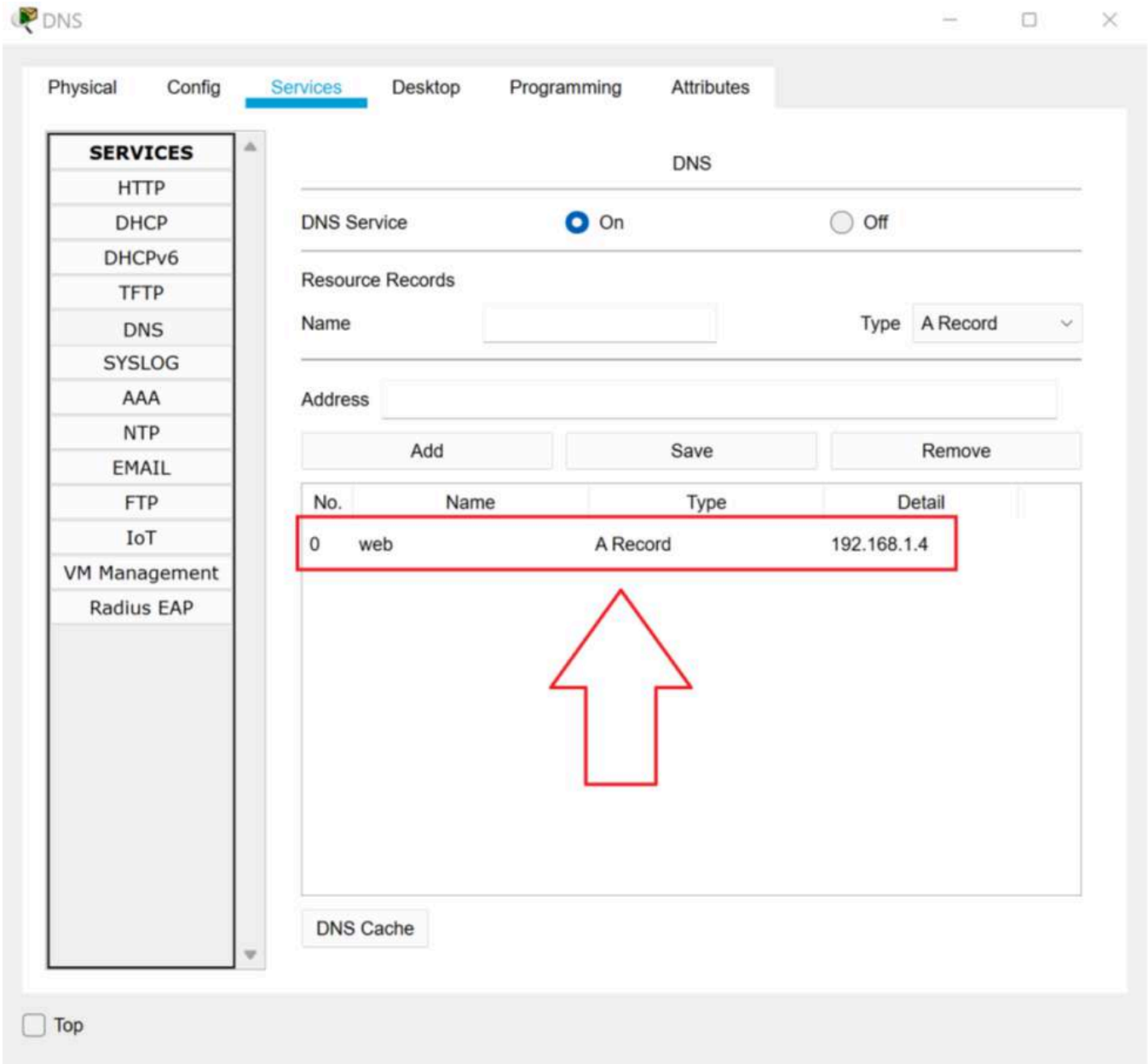


12. In the **Resource Records** section, in the **Name** text box, type **Web**. Verify that the **Type** drop-down combo box is set to **A Record**. In the **Address** text box, type **192.168.1.4**.

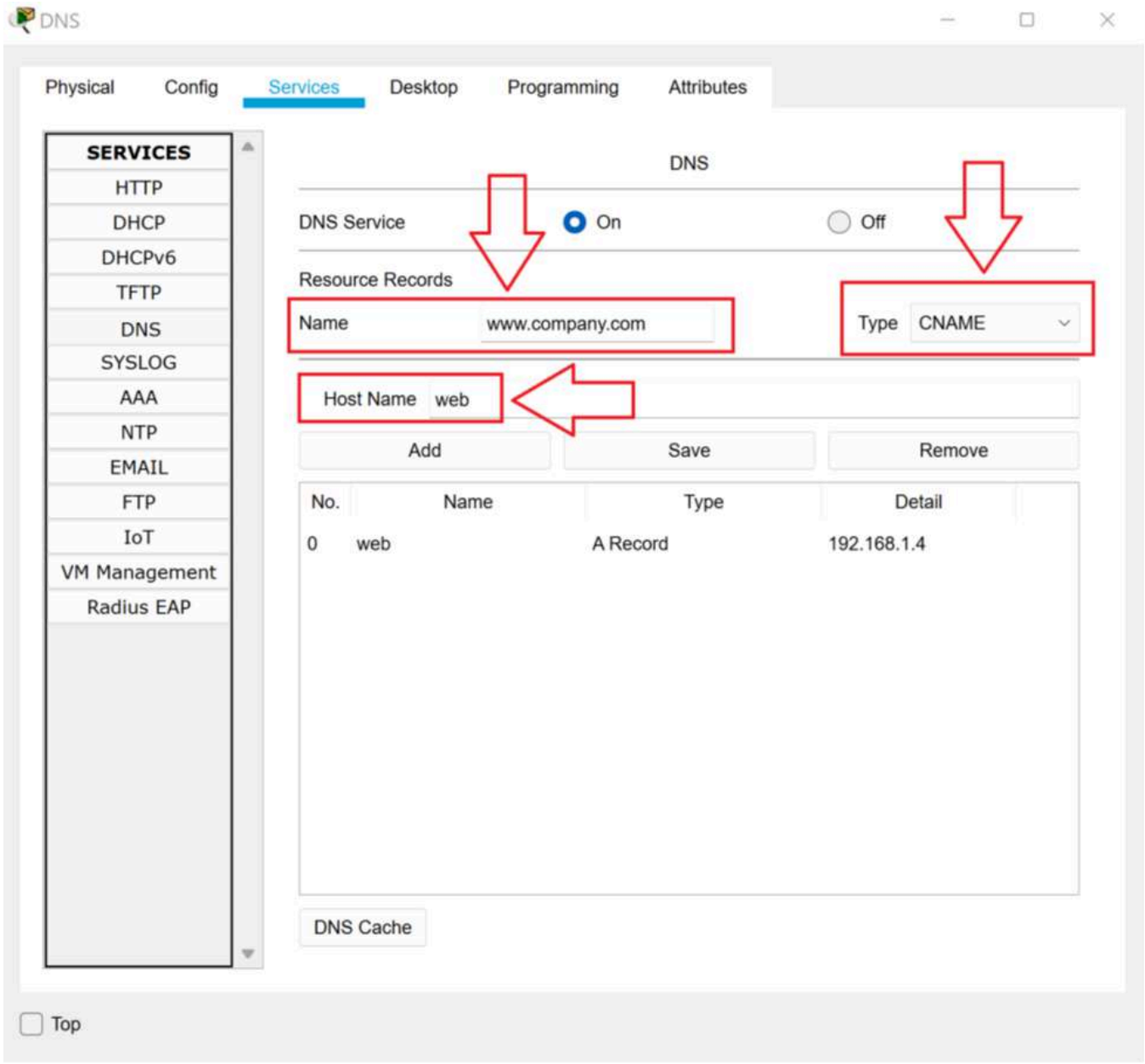


Note: we are creating the A Record without specifying a fully qualified domain since the server does not belong to a domain. Web is the real, only name of this server. An A Record should always reflect the real name of the computer.

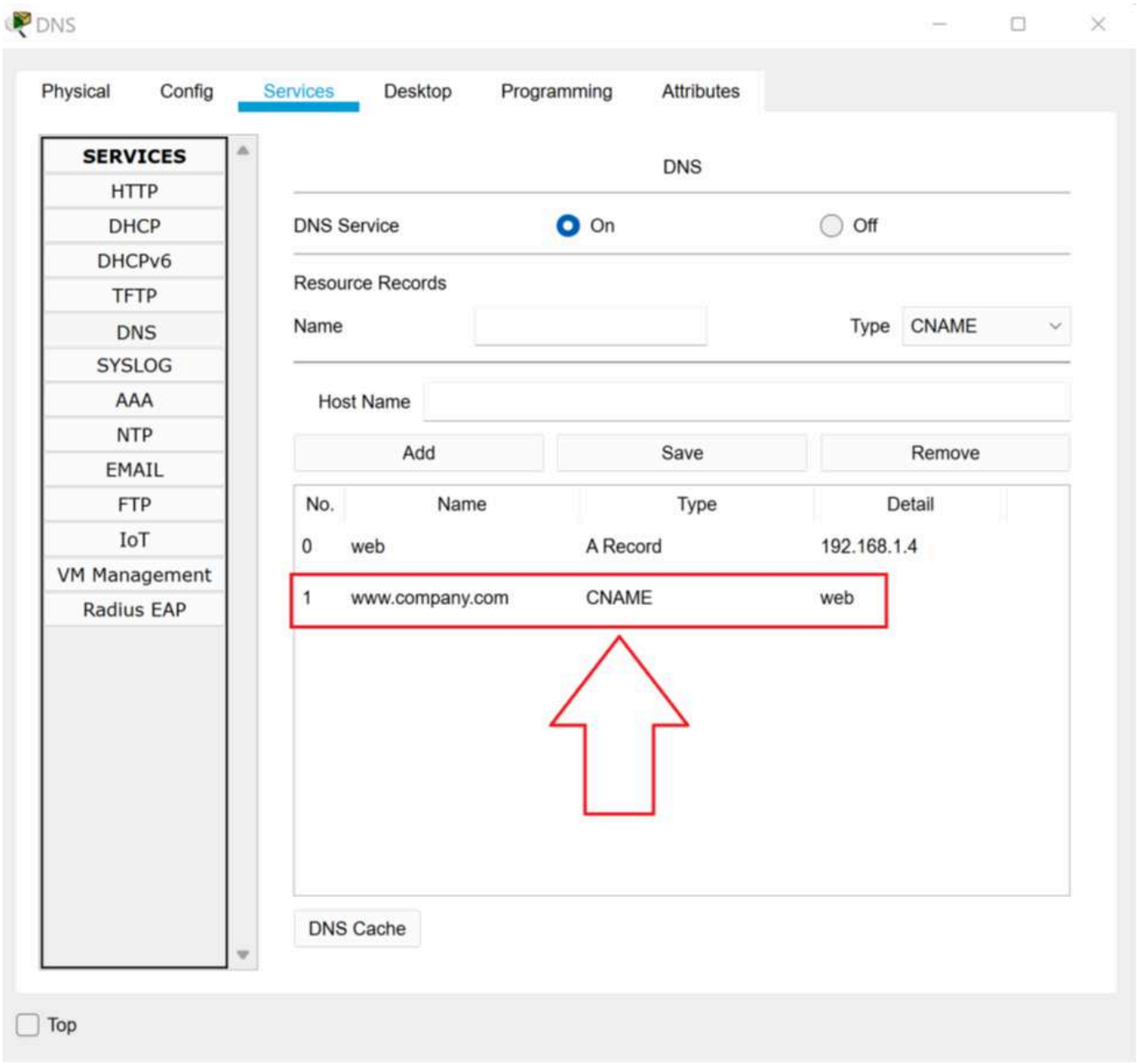
13. Click the **Add** button to create the record.



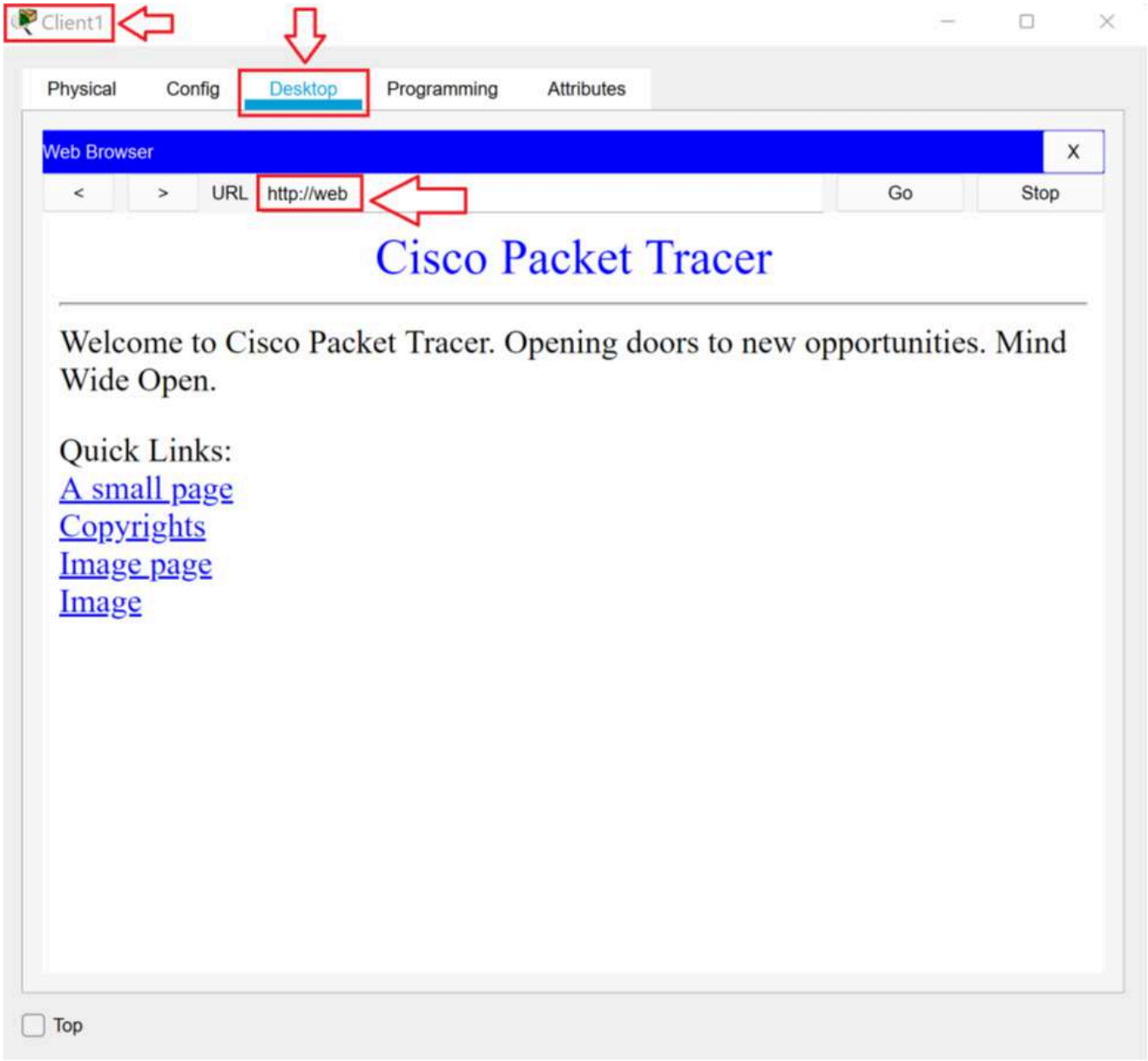
14. In the **Resource Records** section, in the **Name** text box, type **www.company.com**. Change the **Type** drop-down combo box to **CNAME**. In the **Host Name** text box, type **web**.



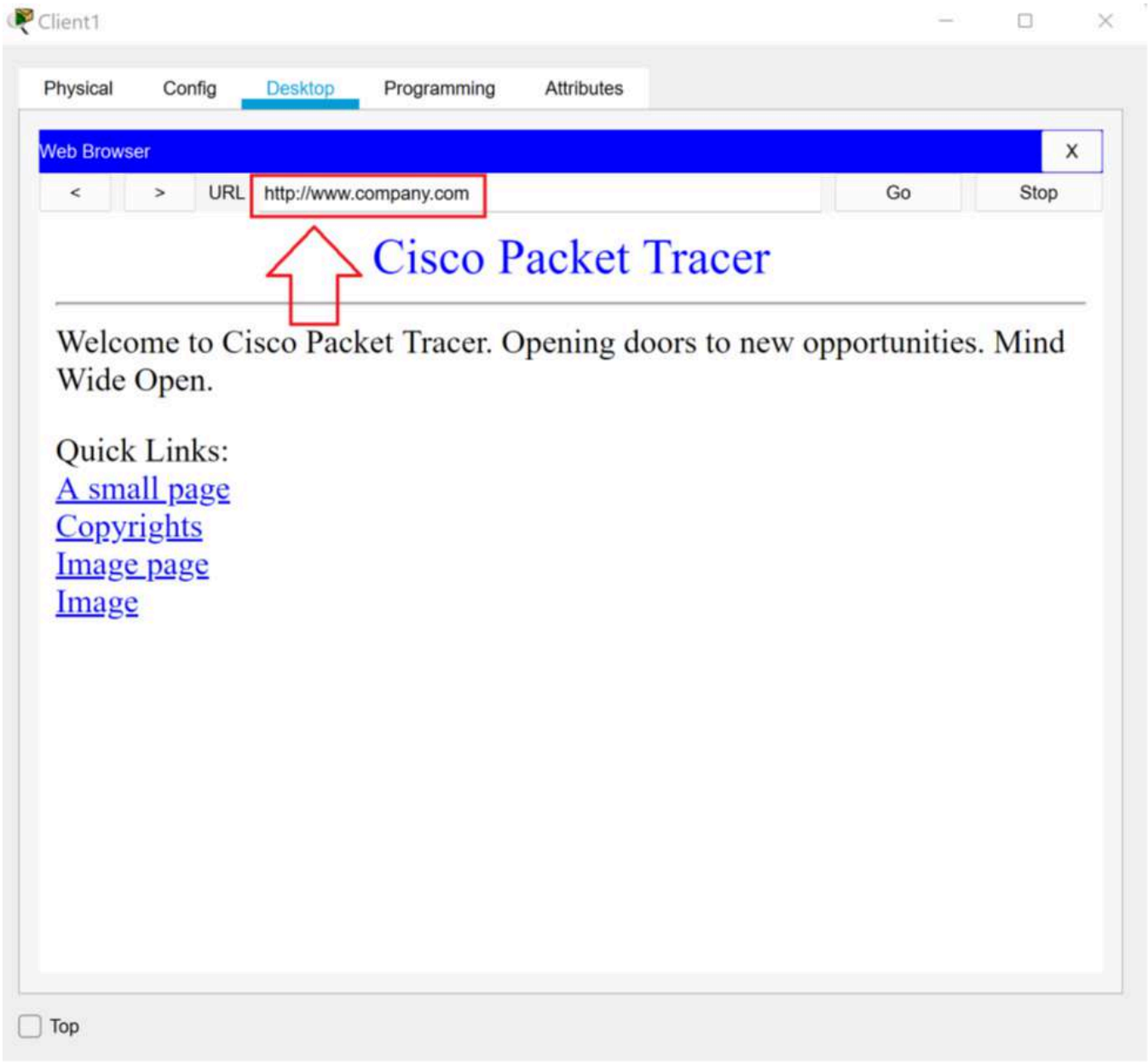
15. Click the **Add** button to create the record.



16. Close the **DNS Properties** dialog box.
17. Click **Client1** to open the **Client1Properties** dialog box. On the **Desktop** tab, click the **WebBrowser** icon. In the **URL** text box, type **http://web** and press **Enter**.



18. In the **URL** text box, type **http://www.company.com** and press **Enter**.

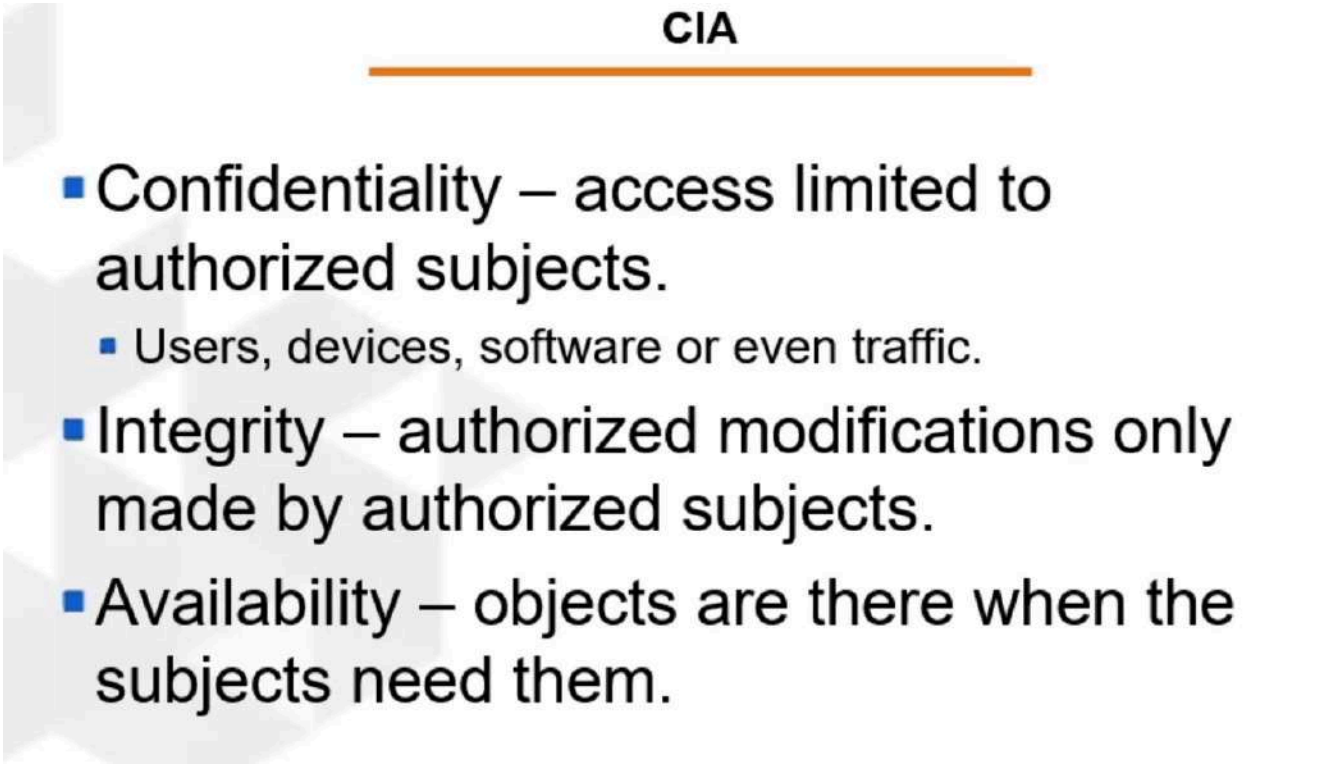


19. Close the **5.4.1 Lab File** file. You do not need to save the changes.

Perimeter Security

Defense in Depth

Security revolves around the CIA triad. CIA stands for confidentiality, integrity, and availability. Confidentiality means that access is limited to authorized subjects. The subjects could be user, devices, software, or even traffic. Integrity means that authorized modifications are only made by authorized subjects. This actually has a couple of things in it. First of all, that the modifications will only be authorized modifications. Even if a subject makes a mistake, we don't let them make an unauthorized modification and that these authorized modifications are only made by authorized subjects. Integrity make sure that what we're supposed to be seeing is the right thing. Then availability means that the objects are there when the subjects need them.

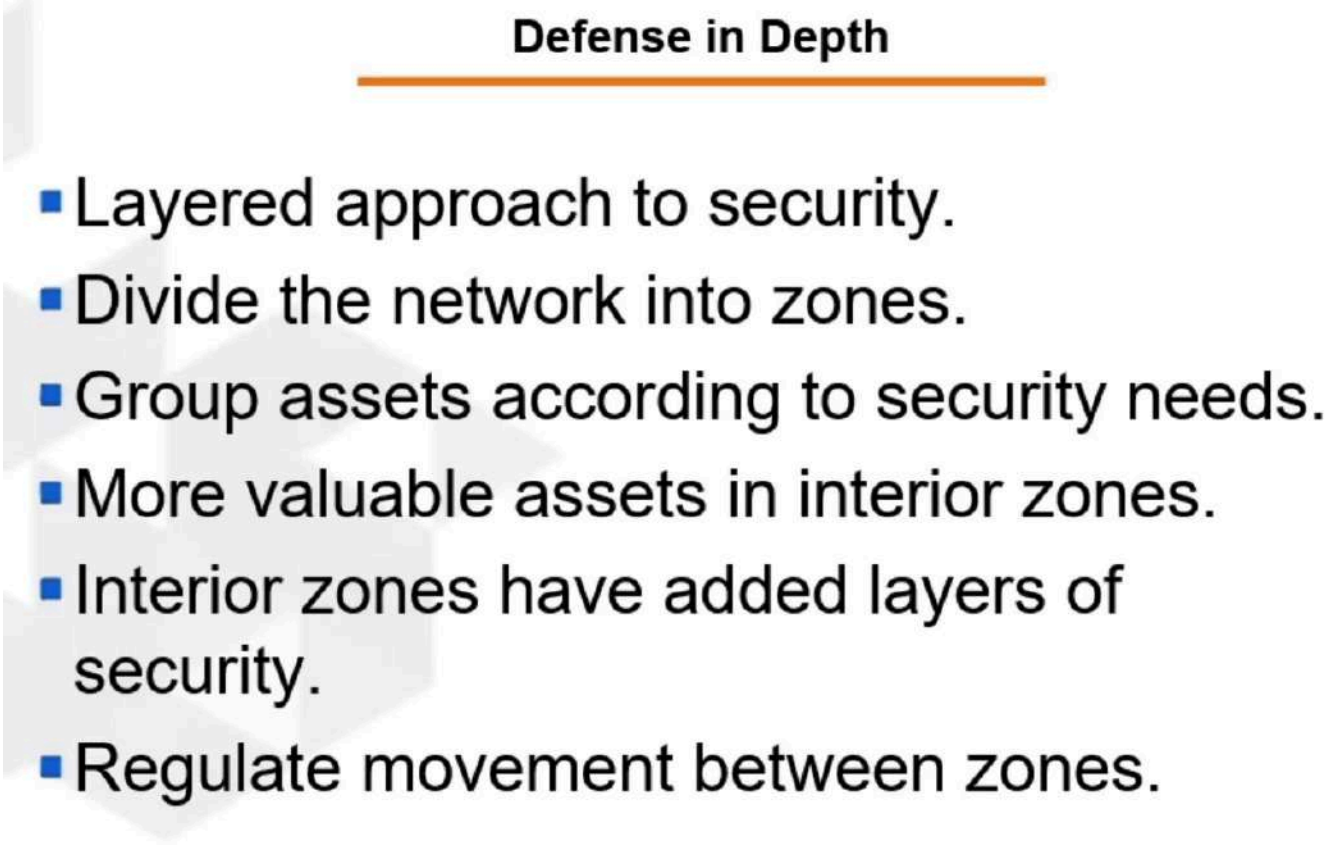


Confidentiality, access is limited only to the people that are supposed to see whatever it is. Integrity, what they're seeing is the right thing and availability, it's there when they need it. All the network security we'll look at in this course relates to confidentiality.

We're going to see several ways to enforce confidentiality, as well as some ways to detect confidentiality violations. Every individual subject and object deserve to be secured appropriately, but it would be difficult and time-consuming to develop a security plan for every individual subject and object.

I mean think about your home. You must have a lot of things in your home that are valuable to you. Think about how much effort it would take to develop a security plan for each of those items. Defense in depth, which is what we use insecurity and I actually use it in my home too, is a layered approach to security.

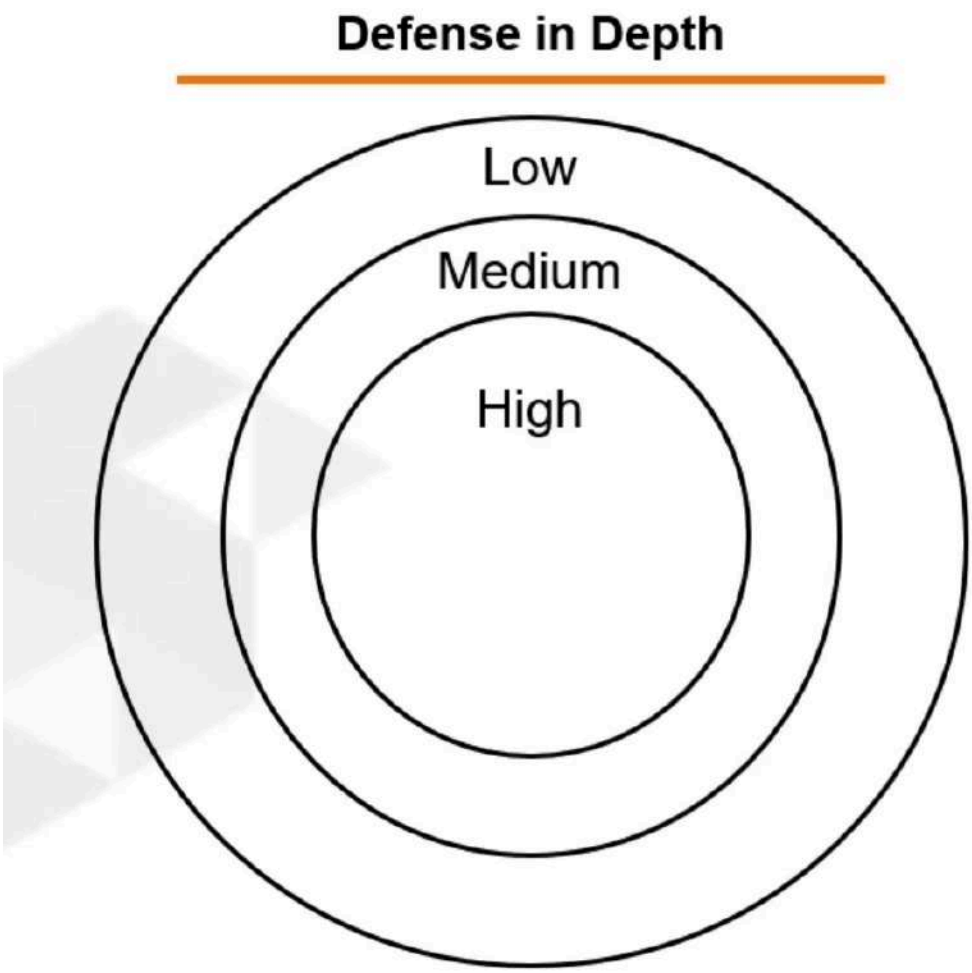
Essentially, what we do is we divide the network into zones, and it doesn't just have to be a network, it could be a physical facility, but you divide whatever it is into zones and then you group your assets according to security needs.



The idea is that the more valuable assets are in the interior zones, and those interior zones have added layers of security. Then what we want to do is regulate movement between the zones. The idea is that as you get closer and closer to the most valuable assets, there are more and more security controls to overcome.

A security control is just anything you do to enforce security. It can be software, it can be hardware, it could even be written policies or training.

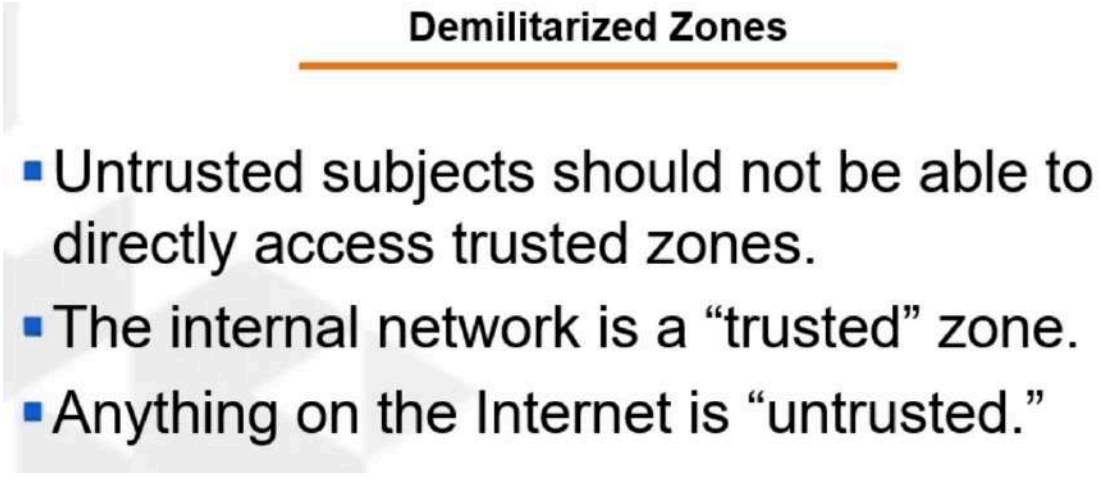
But the idea is that when we divide our assets into zones, we're going to group together assets that have similar security needs, and then we can supply a level of security for the entire zone that matches the security needs of the assets in the zone. We can also regulate entry and exit from the zone.



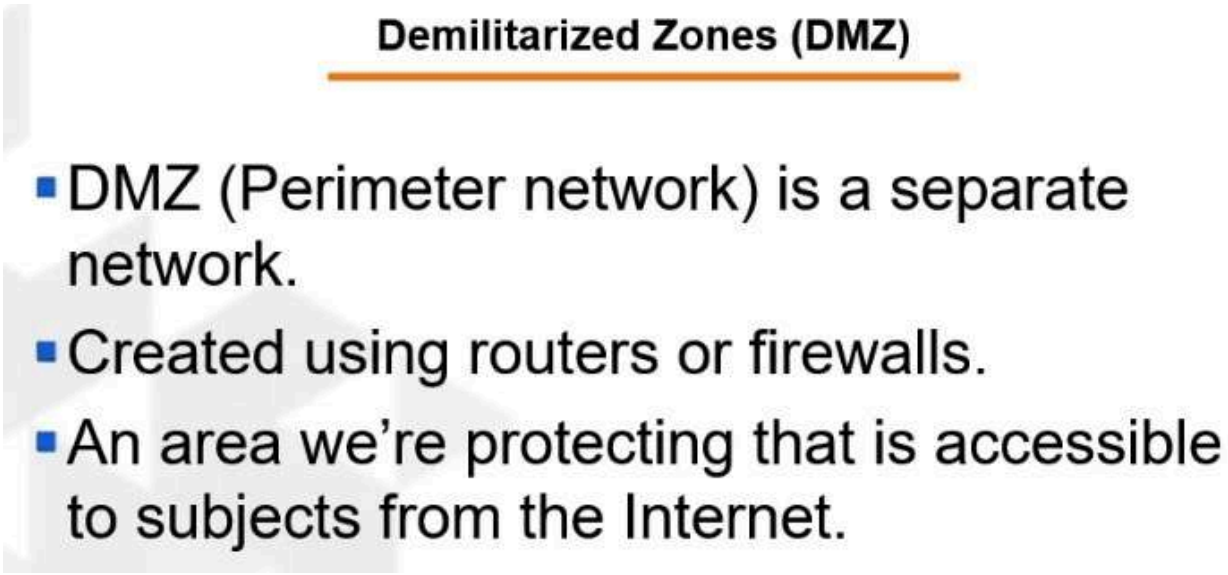
But security is always going to be a divide and conquer type of situation where we're going to organize things into groups and then provide the whole group with a level of security that's reasonable for that group. That's it for this video. In this video we talked about defense in depth which is a layered approach to security for anything we're talking about in the context of networks. We're going to divide our assets into zones. We'll group them together in these zones. We will provide more and more security controls as we move closer and closer to the more valuable assets.

Demilitarized Zones

My home, our bedroom is the zone that has the highest security. When guests arrive, we take their coats in the bedroom and we put them on the bed. When they leave, one of the family retrieves the goat and gives it back to the guest. Wouldn't it be easier just to tell their guests to put their own goat on the bed? Yes, of course. But we don't want even extended family wandering through our bedroom. The same idea it's true in networks. Even if it's for a legitimate reason, untrusted subjects should not be able to directly access objects in a trusted zone. Now, almost all modern networks connect to the Internet and the Internet is a public network. Therefore, the Internet is an untrusted zone. In security and in networking, we refer to the companies or the home's network as the internal network. That means the network inside the company, and the internal network is a trusted zone. Subjects on the internal network can access the Internet. But anything on the Internet is untrusted. We don't want anything untrusted accessing that internal zone.

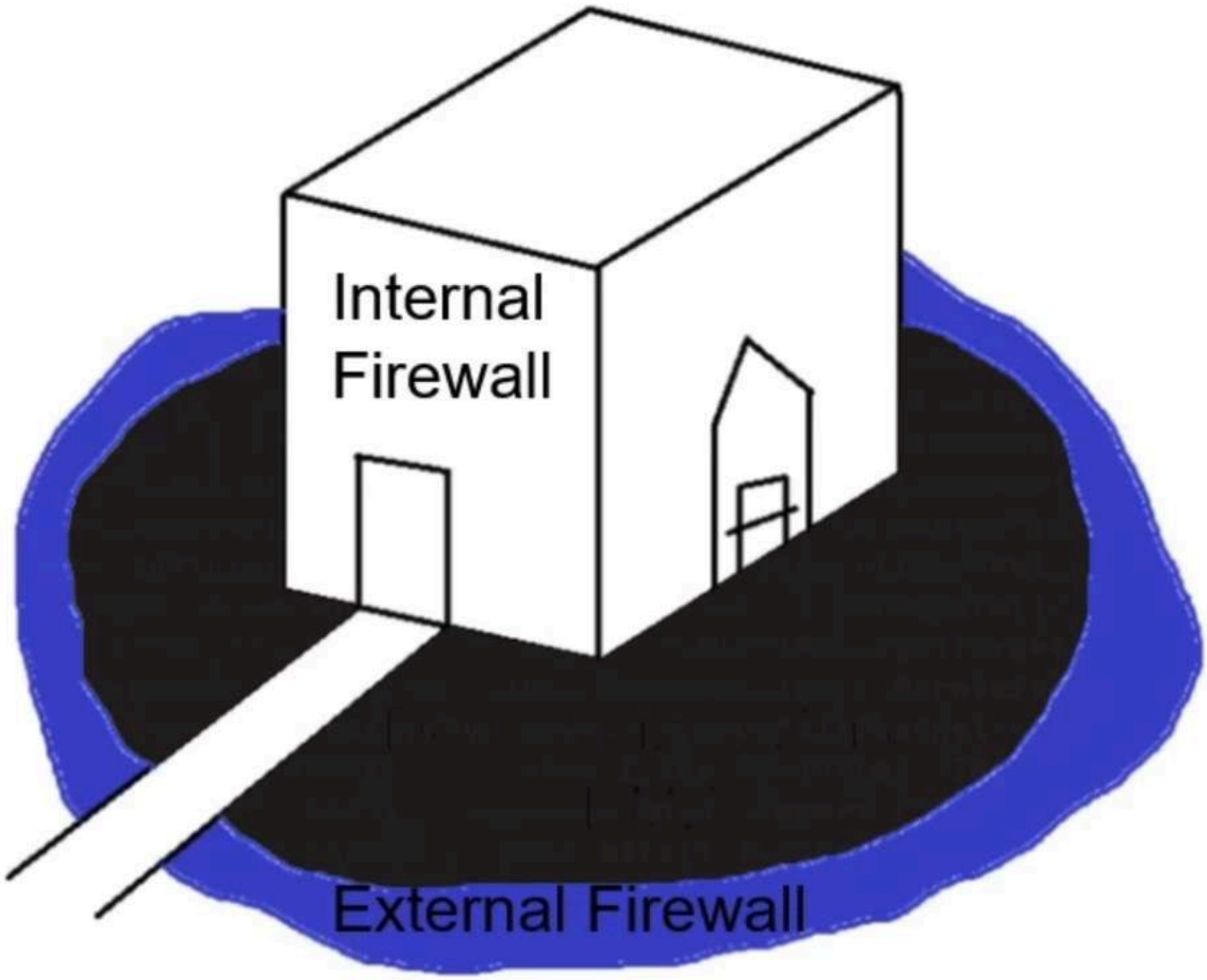


What happens if the company has servers that outside entities need to contact? What if the company has a web server that hosts a website on the internet? What if they have their own mail servers that can receive e-mail from the internet. In that case, the company can place these assets in a demilitarized zone, or DMZ. DMZ might also be called a perimeter network, but it's basically a separate network. It would be created using either routers or firewalls. Essentially, it's an area that we're protecting that's accessible to subject from the Internet now it's also accessible to people from the internal network as well.



But we would put any public meeting accessible on the Internet servers in the DMZ. Now, when I think of DMZ or defense in depth in general, this is the image I always think of and I drew it for you just like I would on a whiteboard if we were in a classroom together, you're thinking, Oh my gosh, that looks like a castle. You're right, that's supposed to be a castle. Remember I teach technology, not bark. Nobody had defense in depth going on like the medieval geeks. The first defense in a castle would be this area here, which is supposed to be water and that's the mode

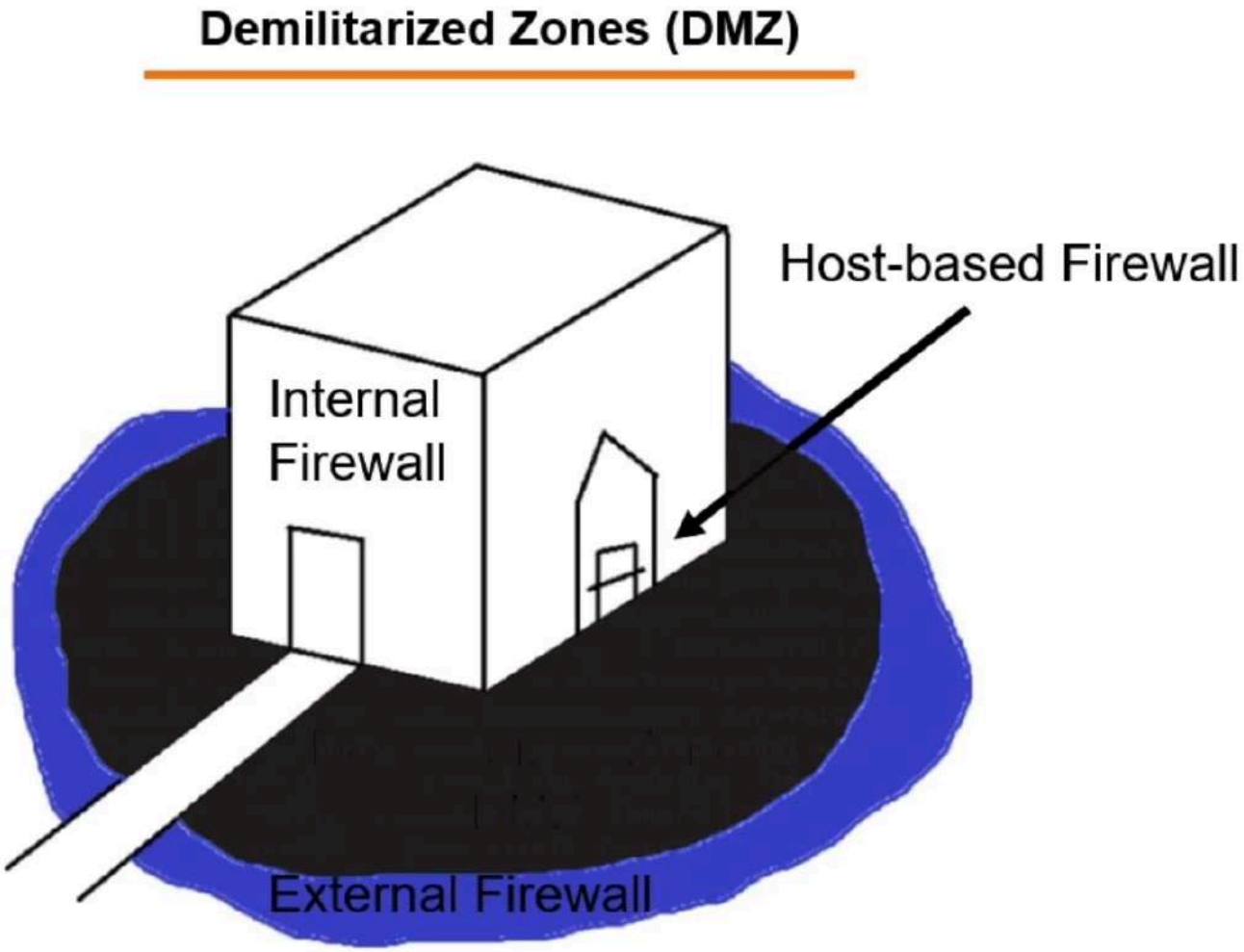
Demilitarized Zones (DMZ)



If we had two firewalls, that would be like the external firewall, the firewall between the DMZ and the Internet. This grassy area in here, that's your DMZ and then the castle walls would be like your internal firewall, the one between the DMZ and the internal network. Then here's your demilitarized zone or DMZ. Put yourself in medieval times.

People are going to come out of the castle walls and be hanging out in the grass selling pottery and swords. People from other castles are going to come across a little drawbridge here. They're going to be buying pottery and swords. It is an area that we're protecting, but we know because outside people can enter this area. We know it's inherently insecure. The internal firewall, the walls of the castle, that we don't allow anybody through that unless they live inside the castle. No, Robin Hood movies where somebody is sneaking in here in disguise. You should not be able to get through that internal firewall unless you're part of the inside network. Now if you notice, I'm not great with drawing, but this is a little house inside the castle

and it's going to bar on the door. That would be analogous to a host-based firewall. Now, why would we still have a bar on the door if it's inside the castle. Well, that's to protect from threats that originate inside the castle or in networking, in the internal network.

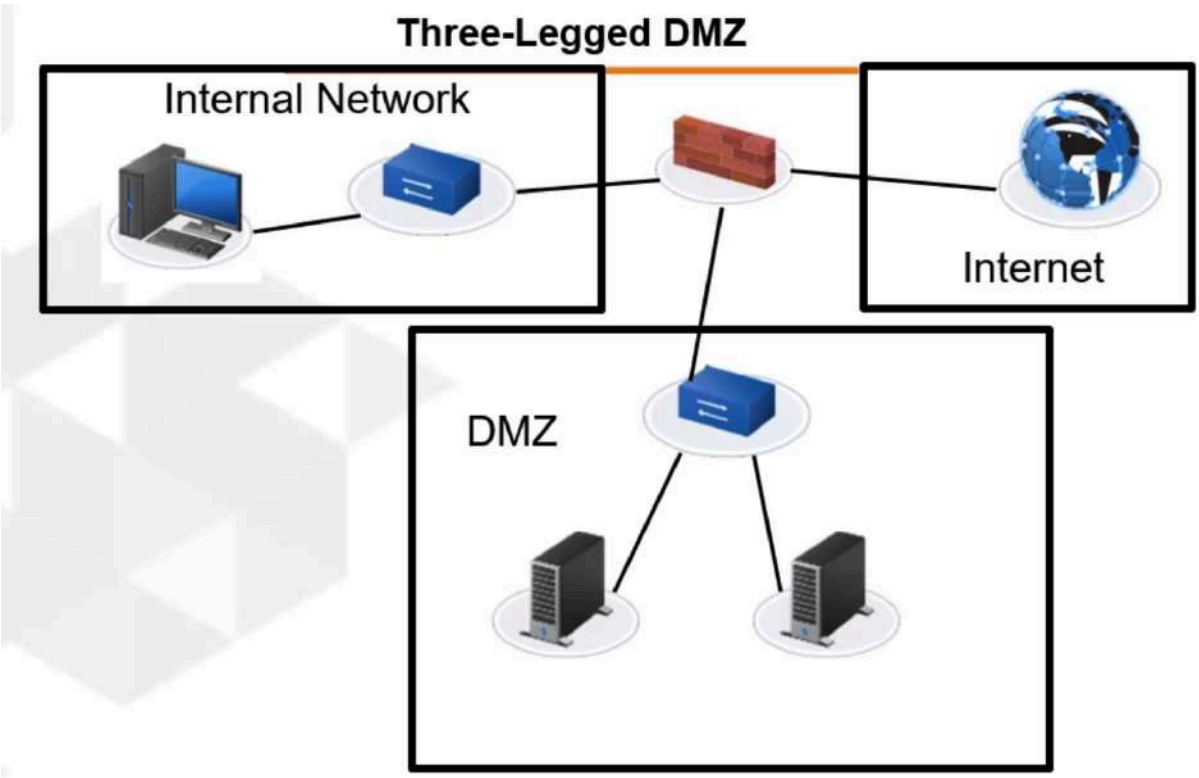


In medieval times you've got picked pockets and people are stealing things. Then there's lots of series you can watch and be very horrified about medieval times. In a network, somebody brings in an infected drive and that starts spreading viruses around the network. We still need protection for the individual items on the internal network. I think this is a really good visual, maybe not the best drawing in the world, but a great visual. If you're trying to think about what is the DMZ? It is an area that we're protecting, but it's inherently insecure because it's exposed to people from the outside world. The DMZ can be accessed by both internal clients, Internet client and you would put anything public, meaning that can be accessed from the Internet in the DMZ.

When somebody says public web server, Republic, this public, that means it can be accessed by people on the Internet. It would go in the DMZ. For applications, gateway also may imply accessible from the Internet.

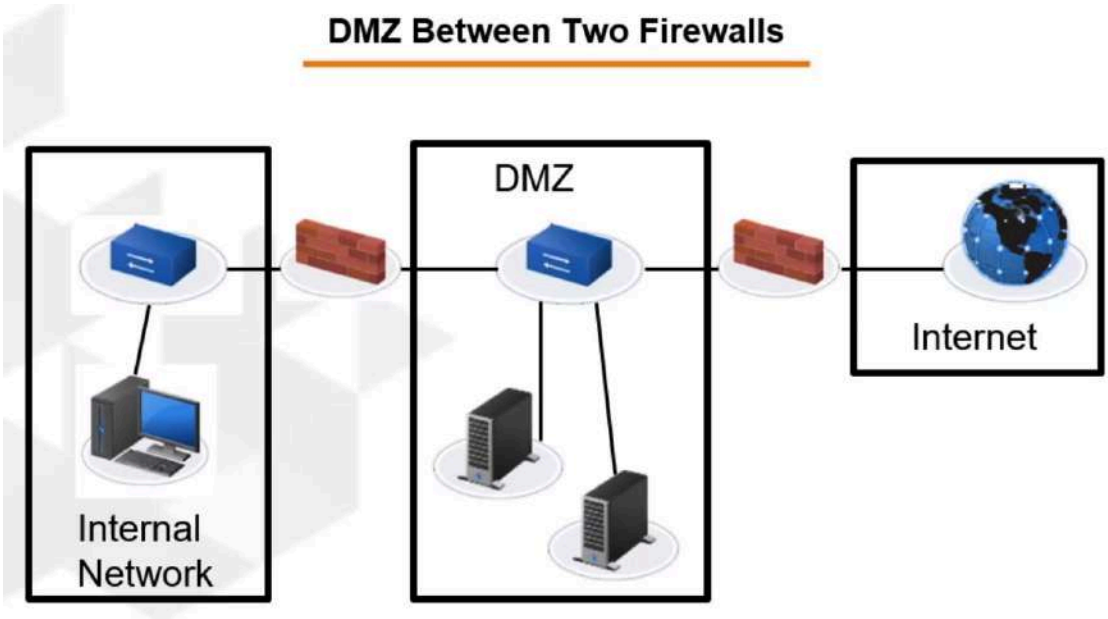
Gateway is a tough term because from a TCPIP perspective, gateway always means a router. We sell the gateway or the default gateway. There's also a definition of Gateway where sometimes when people say Gateway, they mean a device that translates between two incompatible technologies and that's how Routers called gateways, because originally routers were translating between incompatible technologies.Now they usually don't, but it's another definition of that term gateway. Then in terms of software, gateway often means something that accepts data from the Internet. If they said a mail gateway or an email gateway or remote access gateway, in that context, gateway means accessible from the Internet and we would put that in the DMZ.

The two main configurations of DMZ is that you will see, one would be the three-legged DMZ. A three-legged DMZ as you can see here



you have one firewall or router that has three network interface cards. One network interface card that's on the internal network, one that's on the internet, and one that goes to the DMZ. Then you would create rules. In routers we would configure an access control list that would talk about how the traffic is allowed to flow. We would allow outgoing traffic from the internal network to the DMZ into the Internet. We would allow incoming traffic from the Internet to the DMZ, but we would not allow incoming traffic directly from the Internet to the internal network. That would be for a smaller company,

a lot of companies tend to use two firewalls so it would look more like this. They don't really have a term for this. These could be routers or they could be firewalls, and you'd have two of them and the DMZ is in-between them.



In that case, the firewall that's between the internal network and the DMZ would be called the internal firewall. The firewall that's between the DMZ and the Internet would be called the external firewall and that's something I'd want to know if I have to take some assessments a little bit later on. This firewall on the right here, that's the external firewall. This firewall on the left, that's the internal firewall.

DMZ is an area that you're protecting, but that is inherently insecure. It's created using firewalls or routers, and it's where we would put anything that's accessible by clients on the Internet.

Regulating Traffic Between Zones

Once you have divided your network into zones, it’s time to control the flow of traffic between the zones. You can use either a router or a firewall.

Routers

Routers connect two or more different networks and can pass information between them. Routers can be configured with an access control list (ACL) that regulates the flow of traffic between the zones. An access control list is a list configured at an object that controls access to the object. In this case, the objects would be the routers NICs. ACLs on a router typically identify where the traffic is coming from, where the traffic is going to, a protocol, and perhaps a port number. Then you can decide whether to allow or deny the traffic.

Firewalls

Depending on the type of firewall, many firewalls can function like routers. Firewalls are configured with rules. The rules in the firewall would look very similar to the ACL in a router. However, some firewalls allow you to specify a lot more information to use to regulate the traffic. Some firewalls allow you to specify domain names or even key words.

Most firewalls accept incoming traffic and then compare it to a list of rules. The rules are evaluated in order from the top to the bottom. Whichever rule applies to the traffic first is the one that's used. That's why it's important to put specific rules above generic rules.
For example, let's say the rules in a firewall looked like this:

Rule #	Source	Destination	Protocol	Port	Allow/Deny
10	192.168.1.0 /24	192.168.2.40 /24	TCP	80	Deny
20	192.168.1.10 /24	192.168.2.40 /24	TCP	80	Allow

What would be the outcome of 192.168.1.10 sending a message via HTTP to 192.168.2.40?
As the ruleset is configured in the example, that traffic would be denied. But it's clear that the administrator wanted 192.168.1.10 to be an exception to the Deny rule applied to everyone else in the 192.168.1.0 /24 network. In that case, the rules must be ordered like this:

Rule #	Source	Destination	Protocol	Port	Allow/Deny
10	192.168.1.10 /24	192.168.2.40 /24	TCP	80	Allow
20	192.168.1.0 /24	192.168.2.40 /24	TCP	80	Deny

If the rules are properly ordered, the desired outcome is achieved.
Unless otherwise mentioned, you can assume all firewalls have an implicit deny.
An explicit deny is something that is denied by a rule in the firewall. An explicit allow is something that is allowed by a rule in the firewall. The two rules in the example above are both explicit. An "implicit" rule is a setting in the firewall that allows or denies without there being a rule.
A firewall with an implicit allow will allow all traffic unless there is an explicit rule denying the traffic. That is not often used, but sometimes you will see that on outgoing traffic. It's more common for there to be an implicit deny. In that case, all traffic is denied unless there's an explicit rule allowing the traffic.
If you do not have a specific IP address, network ID, protocol or port to specify, you can use the term "Any" to mean "all" in that category.
Sometimes rules can be directional. In that case, they may be written like this:
10 192.168.1.10 /24 > 192.168.2.40 /24 TCP 80 Allow
20 192.168.1.0 /24 <> 192.168.2.40 /24 TCP 80 Allow
When the rule uses a "<" or ">" sign, the sign points in the direction of the destination. A "<>" symbol means the rule applies to traffic going in either direction.

Routers vs Firewalls

If a router with an ACL can act like a firewall, and a firewall can route, why do we need two devices? The device you choose will depend on the primary function the device needs to perform.
Routers can have ACLs. But they have higher level routing functions you wouldn't find in a firewall. Routers can advertise routes to other routers. They also usually have a greater range of the types of media they can use. If your primary goal is to move data through the network in the fastest, most efficient way, you should choose a router.
Firewalls can route, but usually only between their own network cards. They don't "talk" to other routers, and they usually don't have enough routing functionality to evaluate multiple paths through a network to find the best route. On the other hand, with a router you can set up much more complex rules about

which traffic to allow through the router. If your primary goal is to regulate traffic, allowing the traffic you think is harmless and blocking unwanted traffic, you should choose a firewall.

NAT

Sometimes NAT is described as a security feature. NAT replaces the original source address on the packet with the IP address of the public interface on the NAT router. When the reply returns, the NAT router retranslates it to the original sender. However, by default, NAT cannot accept traffic which is not a reply. In this way, it does block some traffic. But it is not a substitute for a firewall.
NAT is a good way to hide details about internal addresses if the internal addresses are public addresses.

Controlling Client Access

Port Security

Besides regulating traffic between zones, it's important to control which clients can connect to the network. The term port security can have several meanings. In this lesson, we mean securing the ports on a switch. If you have an unmanaged switch (one that doesn't have firmware that you can configure), then anyone who plugs into a port on the switch will have access to the network. If you have a managed switch, you can enforce port security.
At a minimum, you should disable unused ports on the switch. The switches are usually stored in LAN closets or telecommunications rooms. Typically, there is a patch panel in the closet or telco room. A patch panel is just a panel that serves as an anchor for the other end of a cable in the wall.
If you imagine an average office with wired connections, there are usually network jacks in the wall. You can plug an ethernet cable into the jack. The other end of the cable plugs into your NIC. Behind the wall jack, there's a cable that runs to a central place which is the closet or LAN room. The wiring in the wall usually belongs to whomever owns the building. But the switches usually belong to whomever rents the facility. If we just put a connector on the other end of the cable coming out of the wall and plugged it into the switch. It would be easy to lose track of which wire goes to which jack on the floor. Instead, the wires are anchored to the patch panel which should be clearly labeled so you know which port in the patch panel connects to which jack in the wall. Then short network cables are run from the patch panel to the switch. (Originally the term "patch cable" just referred to these short cables. Over time, professionals started calling all ethernet cables "patch cables.")
If there is an unused cubical or conference room, that would be a nice place for a hacker to hang out. If it's unused, no one is watching to see what happens. Best practice (the best thing to do) is to disable that port in the switch. Then even if someone plugs a cable into the wall in that area, it won't get them on the network. Of course, if someone needs to use that space, it's going to take some work to get the jack back into production.
The second thing you can do for port security is to enable MAC filtering on the switch. Managed switches often allow you to list out specific MAC addresses and say whether they are allowed or denied connection to the switch. There may be an implicit deny unless you specifically allow the MAC address.
The only problem with MAC filtering is that it's easy to spoof (fake) a MAC address. This is not great security, but it can be an effective way of dealing with rogue devices. In technology, rogue always means unauthorized hardware. By explicitly listing all the MAC addresses that can connect, you are limiting access

to authorized devices. On the other hand, you will have to go find out all the MAC addresses on all the authorized devices.
The last method of port security is to implement 802.1x. 802.1x requires authentication to use the port. When you implement 802.1x, when the client connects to the switch, it will open a login page, usually in a web browser. Someone must log in for the switch to allow traffic from that port to be carried across the switch. This is the best port security, but as you can imagine, requires the most work to implement and support.

Proxy Servers

A proxy server is a system that isolates internal clients from servers on the Internet by downloading and storing files on behalf of the clients. It intercepts requests for web-based or other resources that come from the clients. If it does not have the data in its cache, it can generate a completely new request packet using itself as the source, or simply relay the request. In addition to supplying security, the data cache can also improve client response time and reduce network traffic by supplying often used resources to clients from a local source.
Both proxy servers and NAT devices readdress outgoing packets. However, NAT simply replaces the original source address on the packet. Proxy servers examine the packet contents and then generate a new request packet, thus supplying an added level of protection between the original requesting client and the external network.

Intrusion Detection Systems

Intrusion Detection

Intrusion detection is the process of monitoring the events occurring on a computer or a network and analyzing them to detect incidents. An “incident” is a violation or an imminent threat of violation of both computer security policies and standard security practices. Though this process cannot prevent intrusions from occurring, it is used to monitor events, gather information, create a log of events, and alert you to the incident. The incidents may be unintentional or deliberate, but many of them are malicious. Intrusion detection can be performed manually or automatically.
At its heart, an Intrusion Detection System (IDS) is a sophisticated burglar alarm. Its job is to detect and intrusion, sound an alert and collect information about the intrusion.

Intrusion Prevention

If you want to prevent an intrusion, you must implement an Intrusion Prevention System (IPS). IPSs work exactly like IDSs except that once they detect an intrusion, they take measures to stop it. It might mean automatically changing a firewall rule to block the traffic or directing the traffic to a “black hole” (an IP address or route that leads nowhere but does not inform the sender that the traffic did not arrive.) IPSs may also be called “active IDSs.”
You might be wondering why, if IPSs can prevent intrusions, people use IDSs at all.
An IDS might be part of a layered defense. Suppose the company has a DMZ. You want to detect all the attacks launched at the internal network. You could place an IDS in the DMZ. The firewall between the DMZ and the internal network will filter out some of these attacks. You could place an IPS between the firewall and the DMZ to block any attacks that make it through the firewall. If you did not use an IDS in the DMZ, you would not get information (intelligence) about the attacks blocked by the firewall.

An IDS or an IPS can have a variety of hardware sensors, intrusion detection software, and management software.
There are host-based and network-based products. Host-based products usually are software that run on one host. They protect only that host. You can have Host-based IDSs (HIPS) and Host-based IPSs (HIPS).
Network-based devices scan the network traffic for events that match their rules. You can have Network-based IDSs (NIDS) and Network-based IPSs (NIPS). NIDS can be attached to any port on the switch and configured like a regular packet sniffer. However, in the case of a NIPS, the traffic must flow through the NIPS. When traffic flows through an IDS/IPS, we call it “inline.” If the traffic can flow around the NIPS, the NIPS will not be able to stop intrusions.
The most popular free, open-source IDS/IPS is Snort.

IDS/IPS vs Firewalls

Firewalls regulate traffic based on rules. Traffic is either allowed or denied. IDS/IPS systems detect/prevent intrusions. Generally, intrusions involve patterns in traffic. IDS/IPS systems can regulate traffic according to specific content because they examine packets as they travel through the IPS. It’s more complex than just allowing or blocking traffic. But at heart, these are two different functions. Many devices have services that overlap. Our discussion is more centered around the theory of the devices rather than the implementation.

Types of IDS/IPS

Pattern or Signature Based

An IDS that uses a predefined set of rules that can be pattern-based or uses a signature provided by a software vendor to identify traffic that is unacceptable. Signature-based IDS/IPS systems are only as good as the signature database in the IDS/IPS. As new intrusions are discovered, the vendor adds signatures to the database. These devices should be configured to automatically update.
A signature-based device can only react to known intrusions. Attacks that use vulnerabilities that have not been reported to the vendor, or vulnerabilities the vendor has not yet patched, are called “zero-day attacks.” Signature-based devices cannot block zero-day attacks.

Anomaly or Behavior Based

An “anomaly” is something out of the ordinary. Anomaly-based systems are dynamic. They create a baseline (a picture of “normal” traffic) of acceptable traffic flows during their implementation process. Then they alert when they detect anything out of the ordinary.
Anomaly-based systems can block zero-day attacks. However, they can be difficult to install and configure. During the time the device is learning what is “normal,” they’re prone to false positives.
False positives occur when a security device raises an alarm, but no security incident has occurred. For example, suppose you buy a new computer. The first time you launch Adobe Acrobat Reader (a popular program for reading .pdf files), the anti-virus raises a warning that the software may be a virus. Since this is a legitimate application, the alarm is a false positive.
In addition to false positives, you can also have true positives. A true positive is when a security device raises an alarm and there is a security incident occurring. For example, your computer is infected with a virus. When the anti-virus alerts you, that’s a true positive.
More scary than true/false positives are true/false negatives. A negative occurs when the anti-virus does not alert. For example, if your anti-virus ignores Adobe Acrobat Reader, that’s a true negative. It didn’t alert, and there wasn’t a problem. The worst of all is a false negative. You have a virus, but your anti-virus doesn’t catch it.

During the training period of an IDS/IPS, there may be so many false positives that a true positive isn't flagged. Another issue is that the administrator must check the IDS to make sure the baseline is a good baseline. If the baseline is created when an attack is occurring, the attack may be incorporated into the baseline and go undetected in the future.

Protocol Based

A protocol-based IDS/IPS might be installed on a web server and used to check the protocol(s) used by the computer. It has a system or agent at the front end of a server that is used for the monitoring and analysis of the communication protocol between a connected device and the system. They are also dynamic devices that create a baseline and then look for anomalies in how the protocols are based.

Application Protocol Based

An application protocol-based IDS/IPS that checks the application protocol(s) in use by the system. It has an agent that interfaces between a process, or between multiple servers and analyzes the application protocol between two devices. Application-based devices check traffic within or related to a specific application. They may be used in conjunction with a network- or host-based device to add another layer of protection to a critical application, such as a customer database. They are also dynamic devices that create a baseline and then look for anomalies in how the protocols are based.

Network Security Lab

Exploring NAT

In this lab you will configure NAT to hide the details of internal and DMZ addresses.

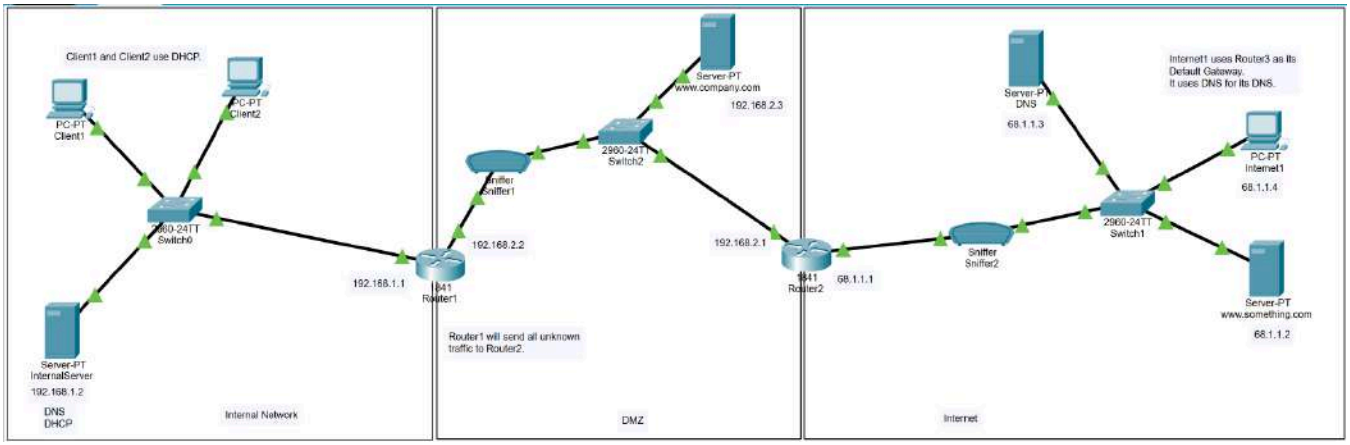
TASK A

In this task, you will test connectivity to the web server in the DMZ.

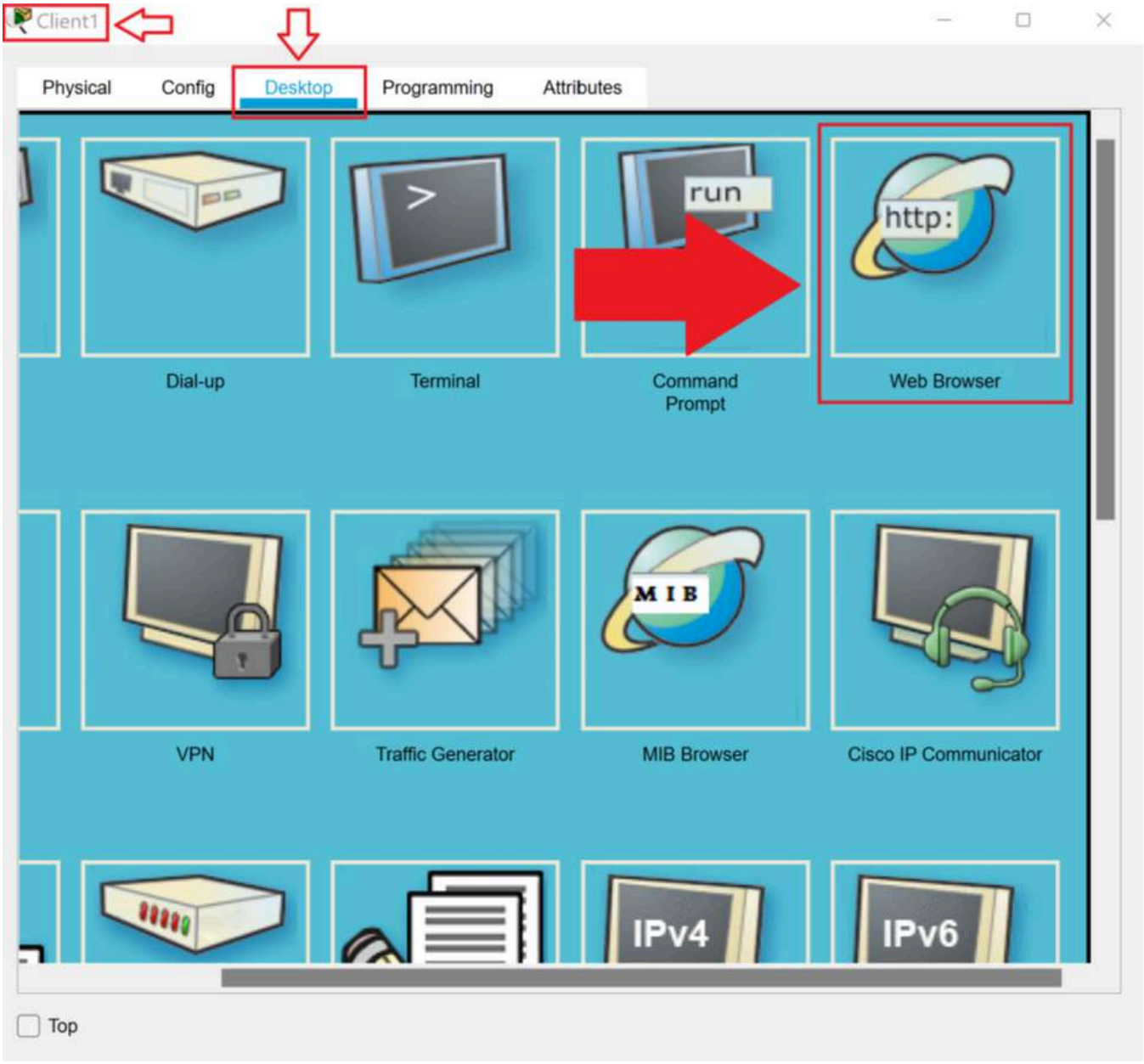
1. Download the **6.3.1 Lab File** and open it in **Packet Tracer**.

6.3.1 Lab File

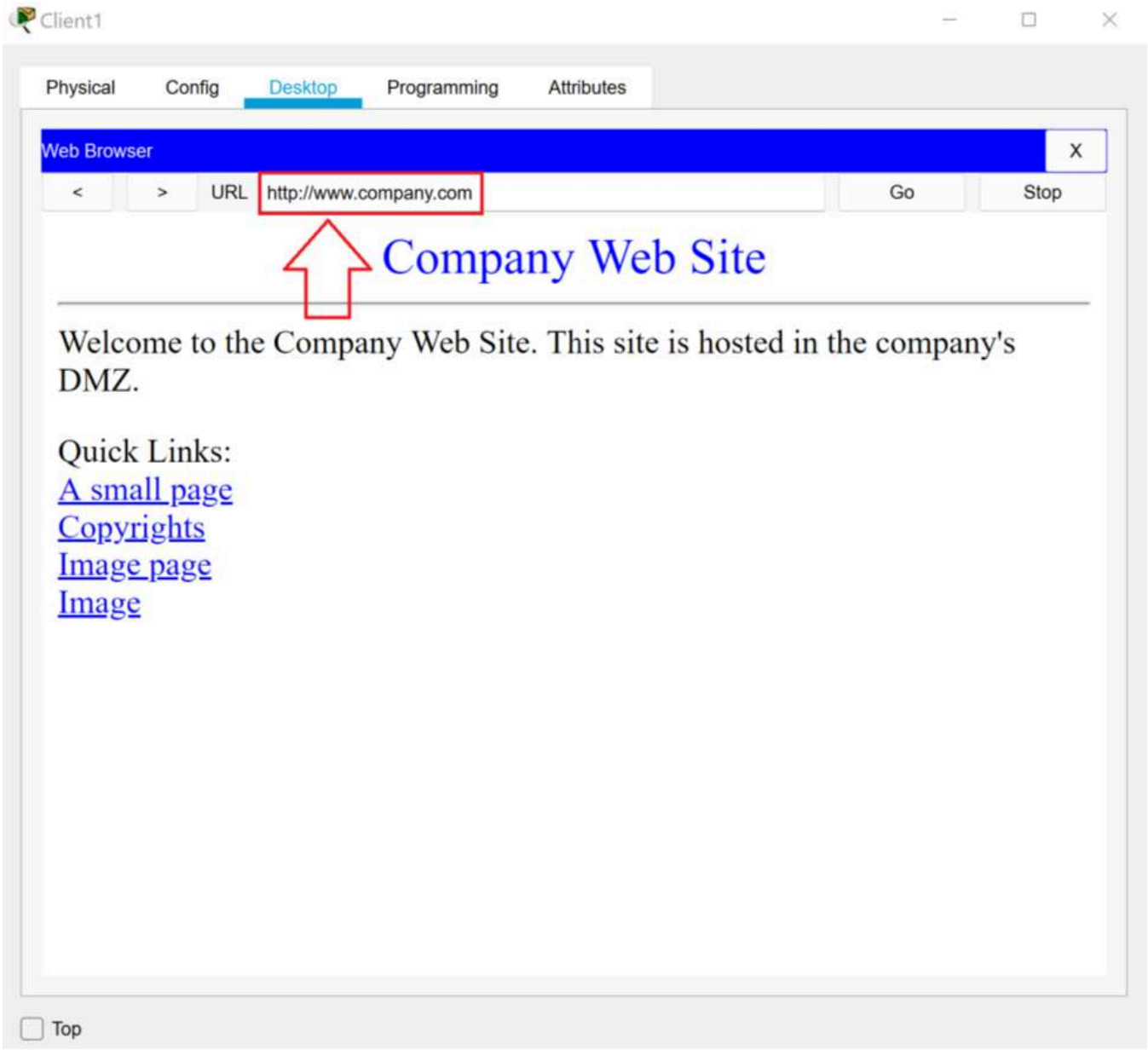
PKT File



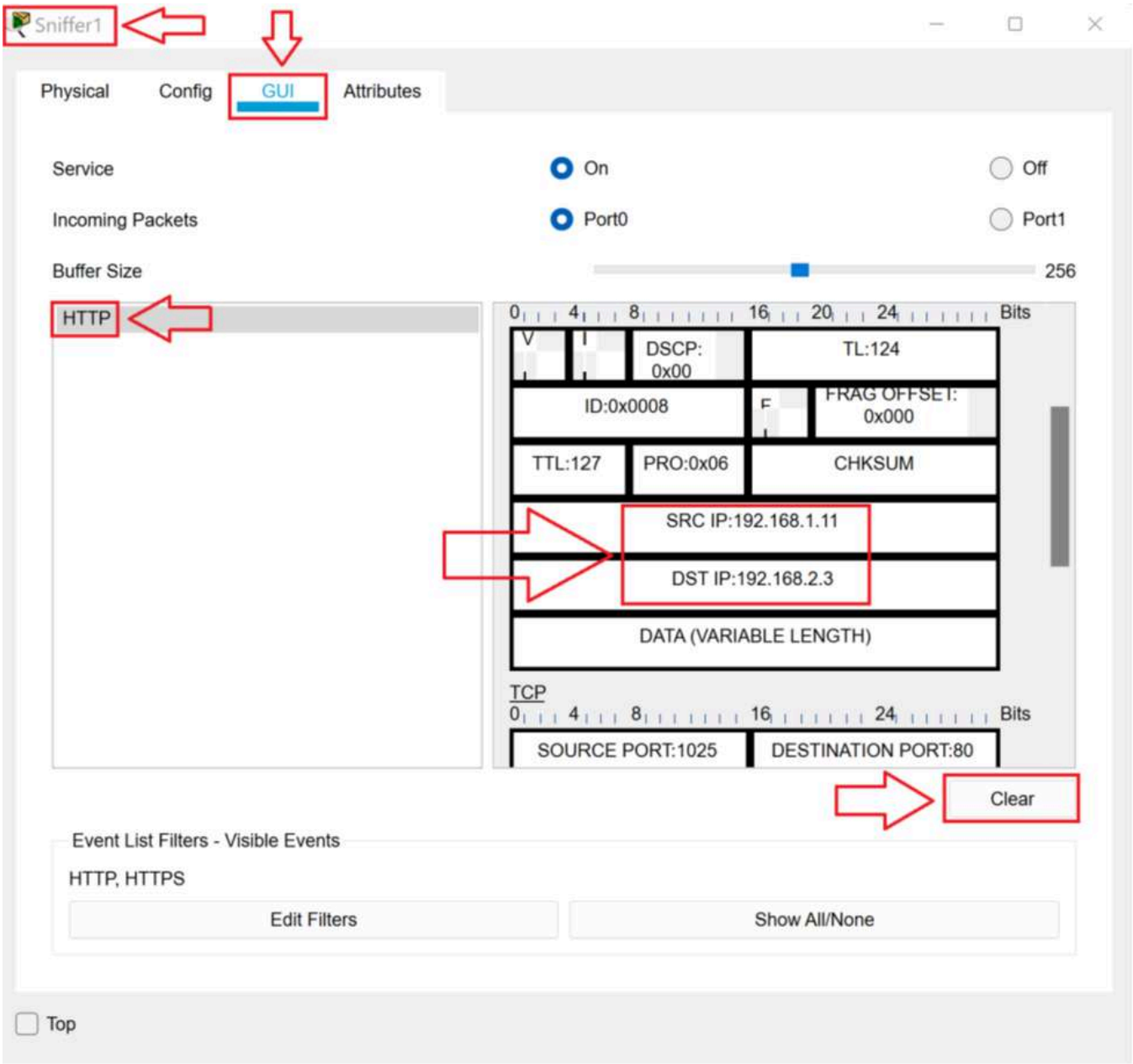
2. Click **Client1** to open the **Client1 Properties** dialog box. On the **Desktop** tab, scroll to the right and then click the **Web Browser** icon.



3. In the **URL** text box, type **www.company.com** and press **Enter**.

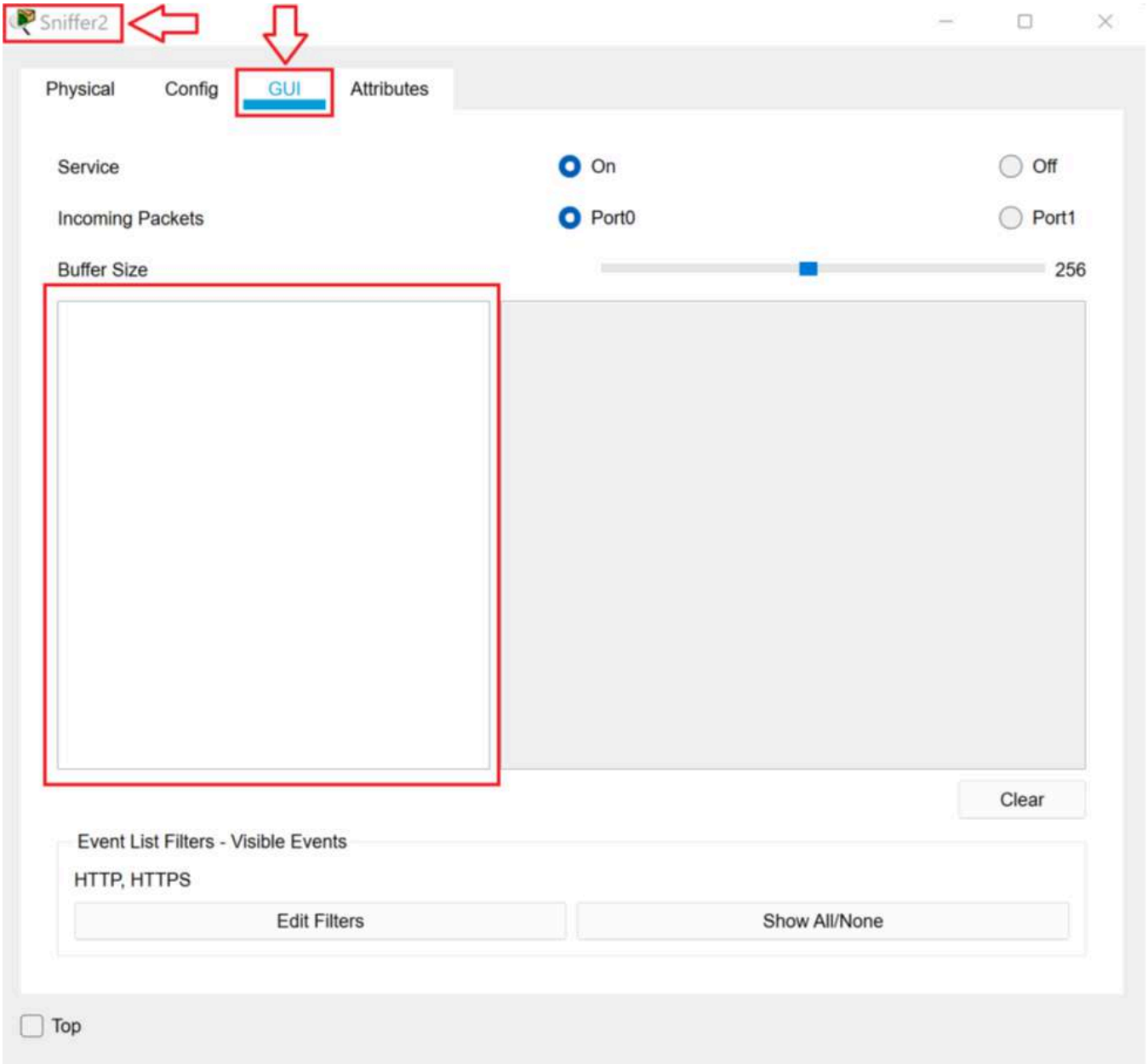


4. Close the **Client1 Properties** dialog box.
5. Click **Sniffer1** to open the **Sniffer1 Properties** dialog box. On the **GUI** tab, select the **HTML** packet. Notice that the source (**SRC IP**) is the source IP address of Client1 and the destination address (**DST IP**) is the address of www.company.com.Click the **Clear** button and then close the **Sniffer1 Properties** dialog box.



Click the **Clear** button.

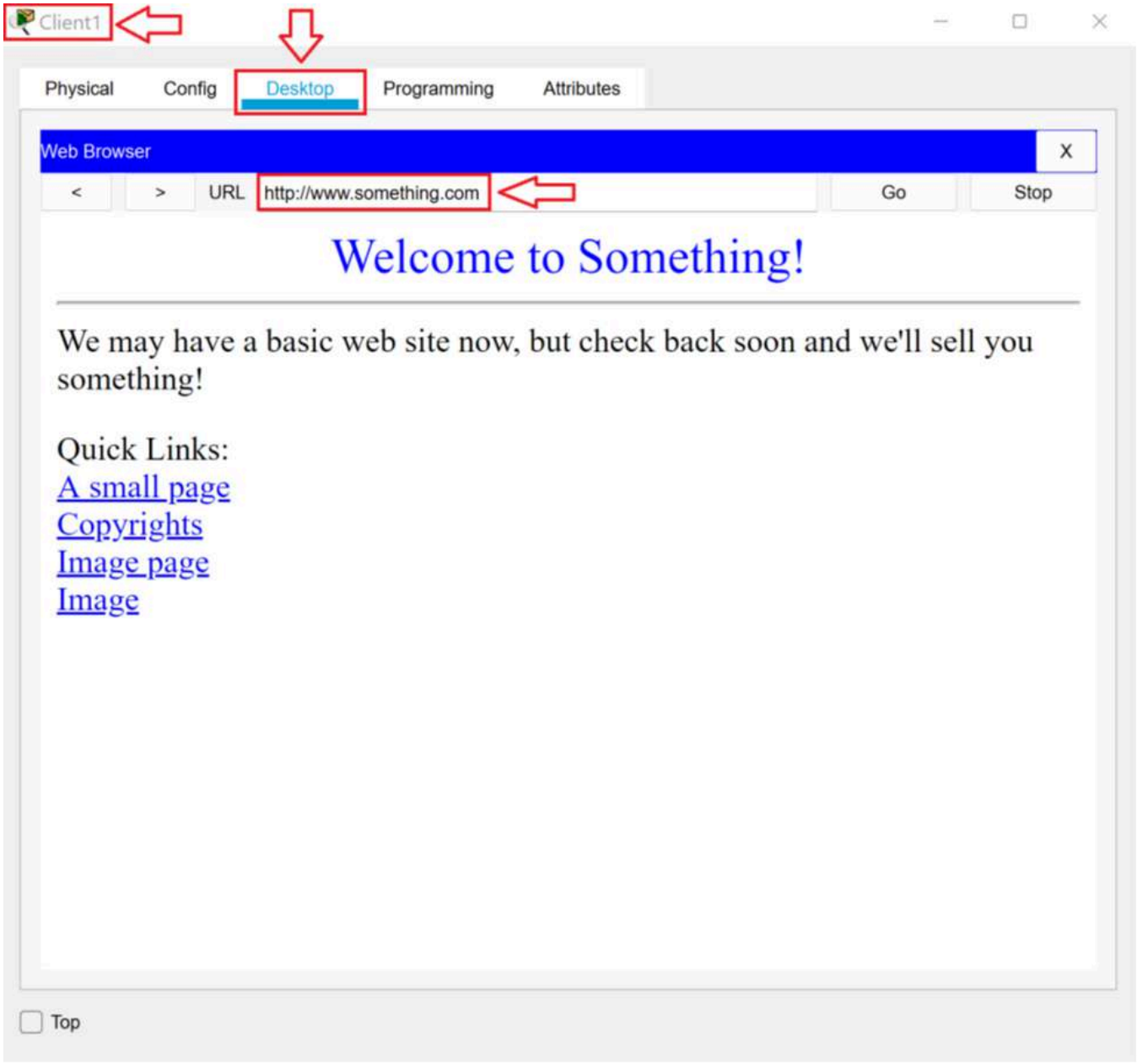
6. Click **Sniffer2** to open the **Sniffer2 Properties** dialog box. On the **GUI** tab, verify that Sniffer2 has not recorded any traffic.



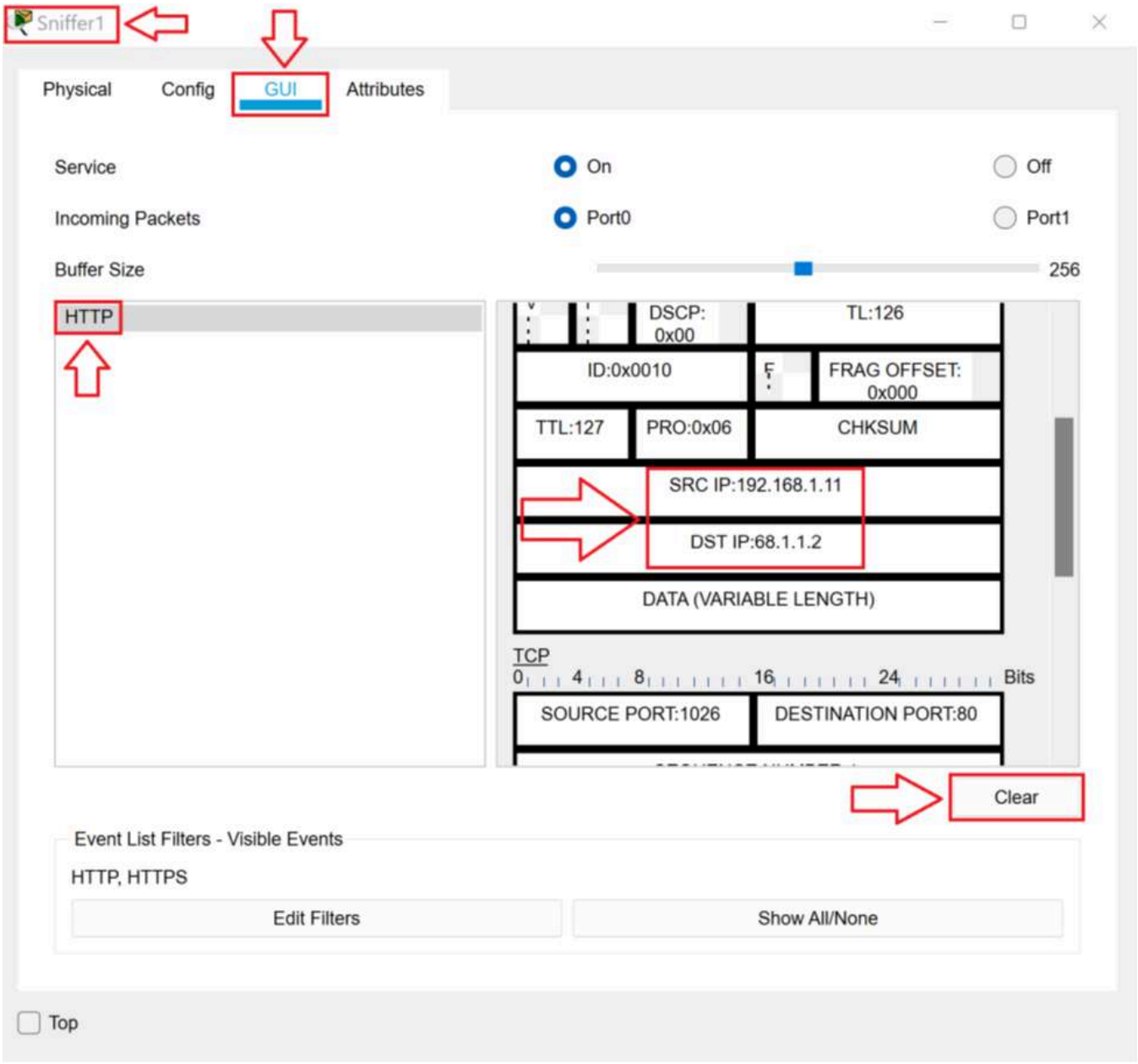
TASK B

In this task, you will test connectivity to a web server on the Internet.

1. Click **Client1** to open the **Client1 Properties** dialog box. On the **Desktop** tab, in the **Web Browser**, in the **URL** text box, type **www.something.co**m and press **Enter**.



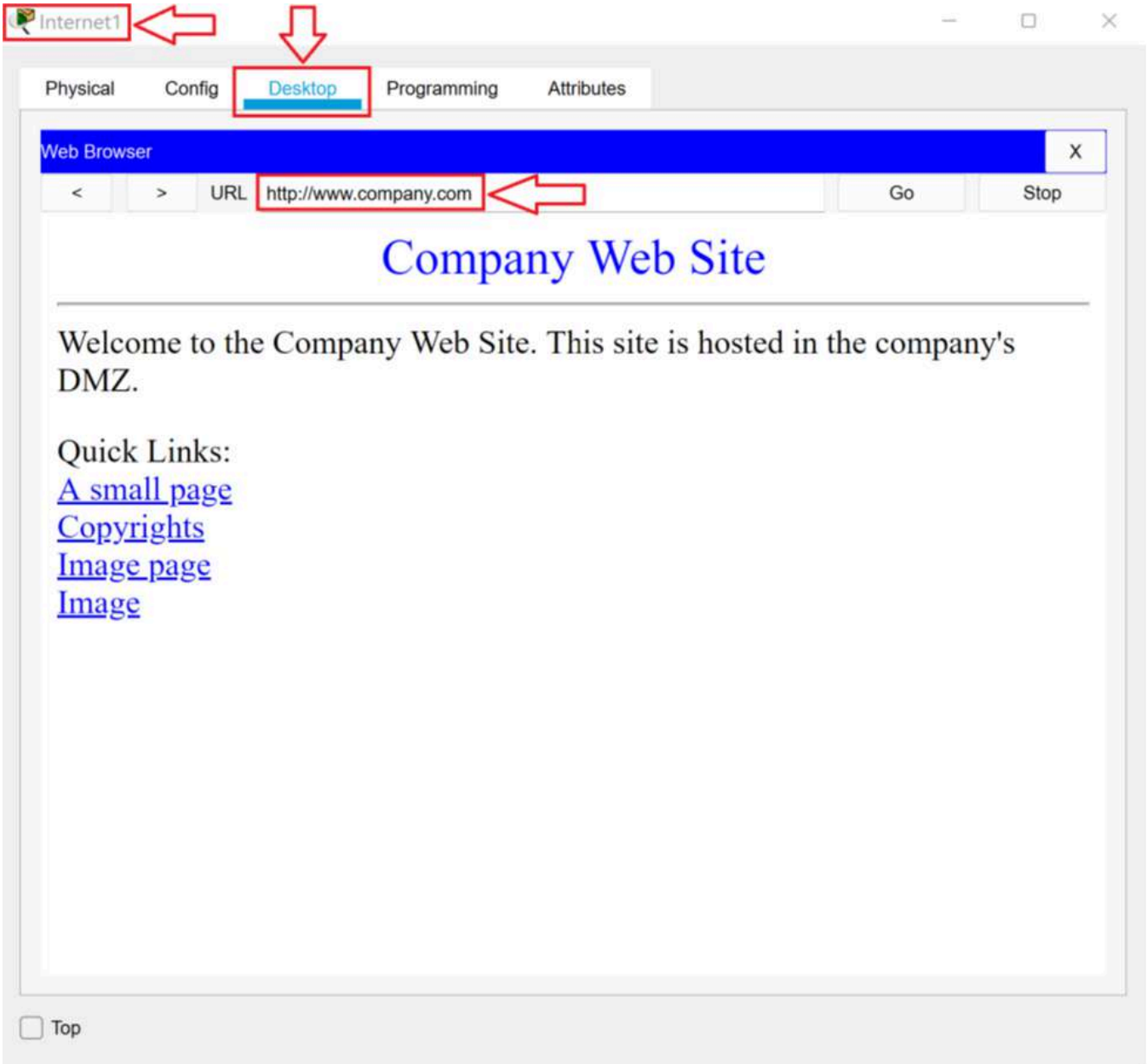
2. Close the **Client1 Properties** dialog box.
3. Click **Sniffer1** to open the **Sniffer1 dialog** box. On the **GUI** tab, select the **HTML** packet. Notice that the source is the source IP address of Client1 and the destination address is the address of www.something.com.Click the **Clear** button and then close the **Sniffer1 Properties** dialog box.



TASK C

In this task, you will test Internet connectivity to the company web server.

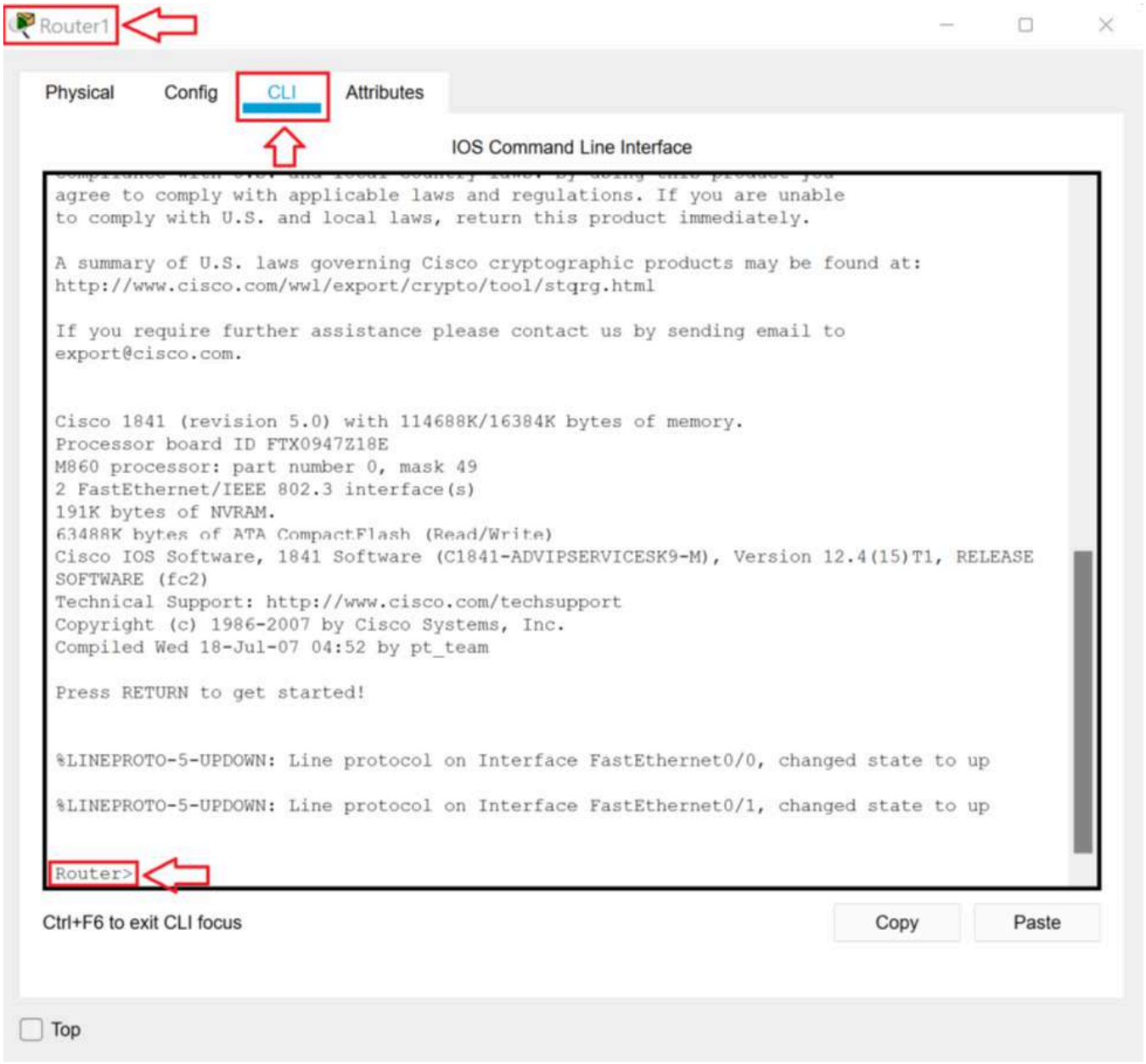
1. Click **Internet1** to open the **Internet1 Properties** dialog box. On the **Desktop** tab, in the **Web Browser**, in the **URL** text box, type **www.company.com** and press **Enter**. Notice that the Internet client can access the company web site.



TASK D

In this task, you will configure Router1 for NAT.

1. Click **Router1** to open the **Router1 Properti**es dialog box. On the **CLI** tab, click anywhere inside the box and then press **Enter**. (The prompt will change to **Router>**)



2. Type **enable** and press **Enter**.

```
Router>enable
Router#
```

3. Type **config terminal** and press **Enter**.

```
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
```

4. Type **int Fa0/0** and press **Enter**.

```
Router(config)#int Fa0/0
Router(config-if)#
```

5. Type **ipnat inside** and press **Enter**. This tells the router that FastEthernet0/0 is on the inside network.

```
Router(config-if)#ip nat inside
Router(config-if)#
```

6. Type **int Fa0/1** and press **Enter**.

```
Router(config-if)#int Fa0/1
Router(config-if)#
```

7. Type **ipnat outside** and press **Enter**. This tells the router that FastEthernet0/1 will be the outside (public) adapter.

```
Router(config-if)#ip nat outside
Router(config-if)#
```

8. Now that you have identified the two interfaces, you must set up the filter that allows traffic in the right direction. Type **access-list 1 permit 192.168.1.0 0.0.0.255** and press **Enter**. This creates an access list permitting all IP addresses from the internal network. Note: There is a space between 192.168.1.0 and 0.0.0.255 which identifies all hosts.

```
Router(config-if)#access-list 1 permit 192.168.1.0 0.0.0.255
Router(config)#
```

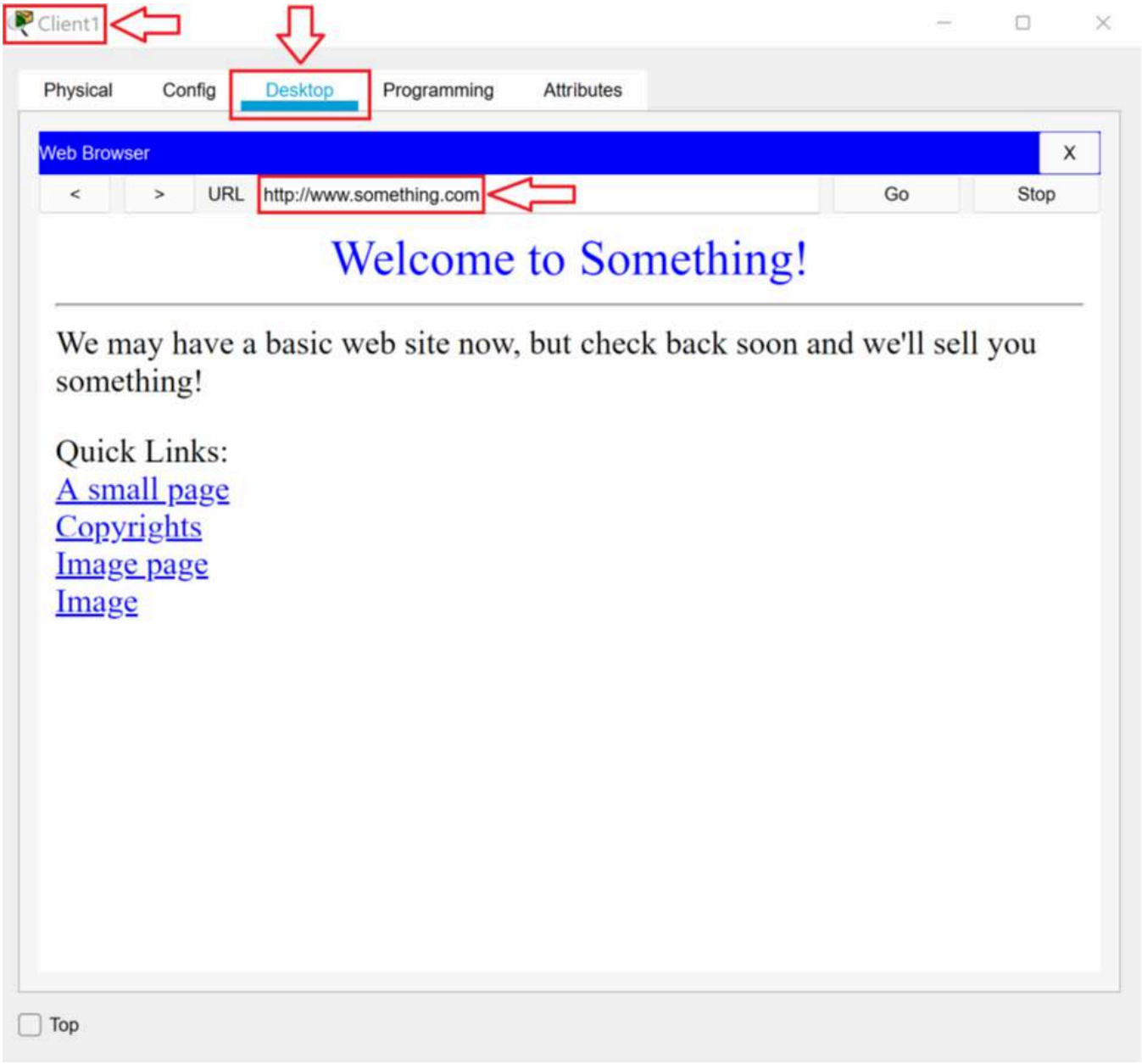
9. The final command enables NAT. Type **ipnat inside source list 1 interface Fa0/1 overload** and press **Enter**.

```
Router(config)#ip nat inside source list 1 interface Fa0/1 overload
Router(config)#
```

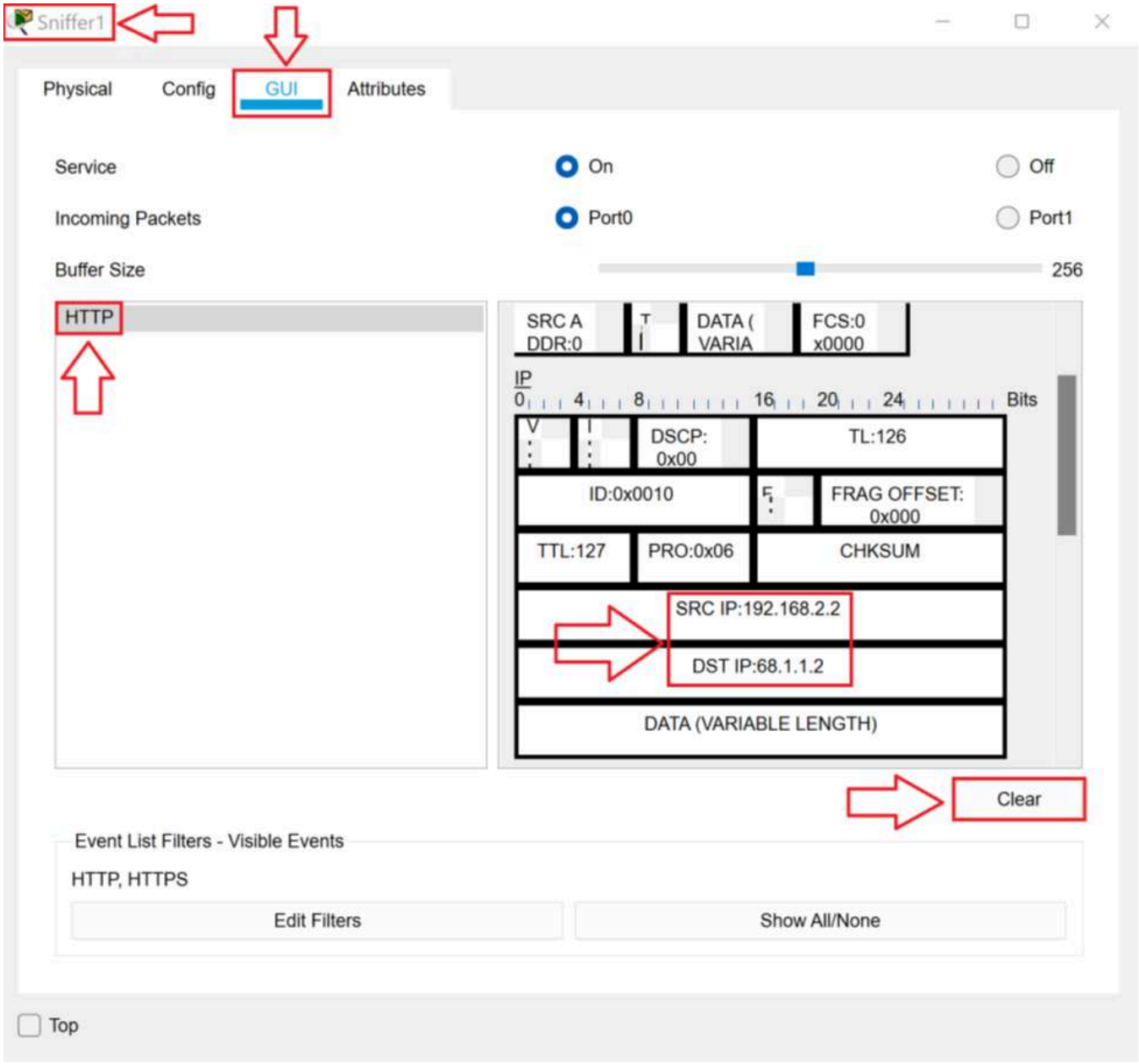
TASK E

In this task you will test NAT on Router1.

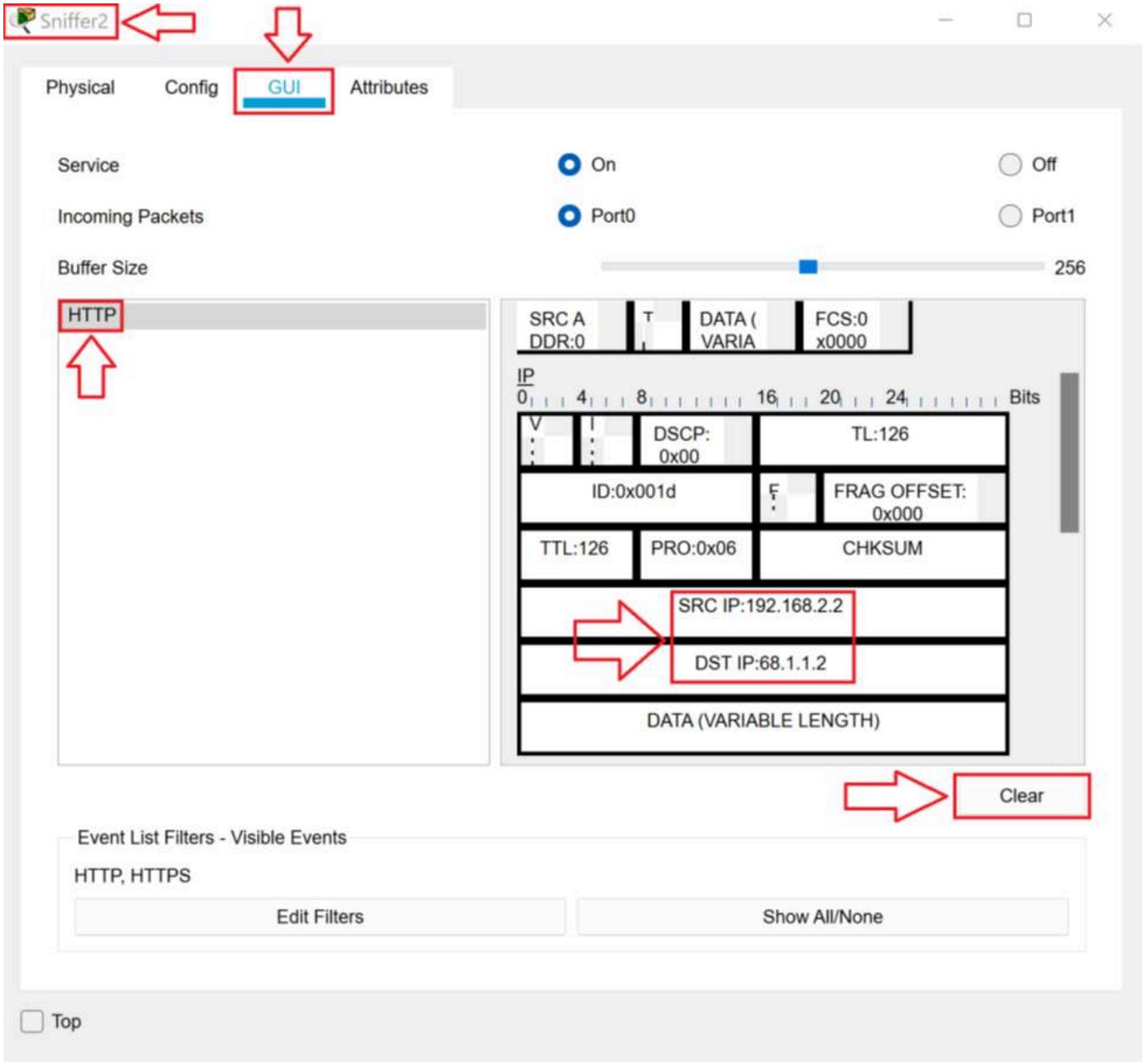
1. Click **Client1** to open the **Client1 Properties** dialog box. On the **Desktop** tab, in the **WebBrowser**, in the **URL** text box, type **www.something.com** and press **Enter**. Click **Clear** and then close the **Client1 Properties** dialog box.



2. Click **Sniffer1** to open the **Sniffer1 Properties** dialog box. On the **GUI** tab, select the **HTML** packet. Notice that the source is the source IP address of Router1 and the destination address is the address of www.something.com. Click the **Clear** button and then close the **Sniffer1 Properties** dialog box.



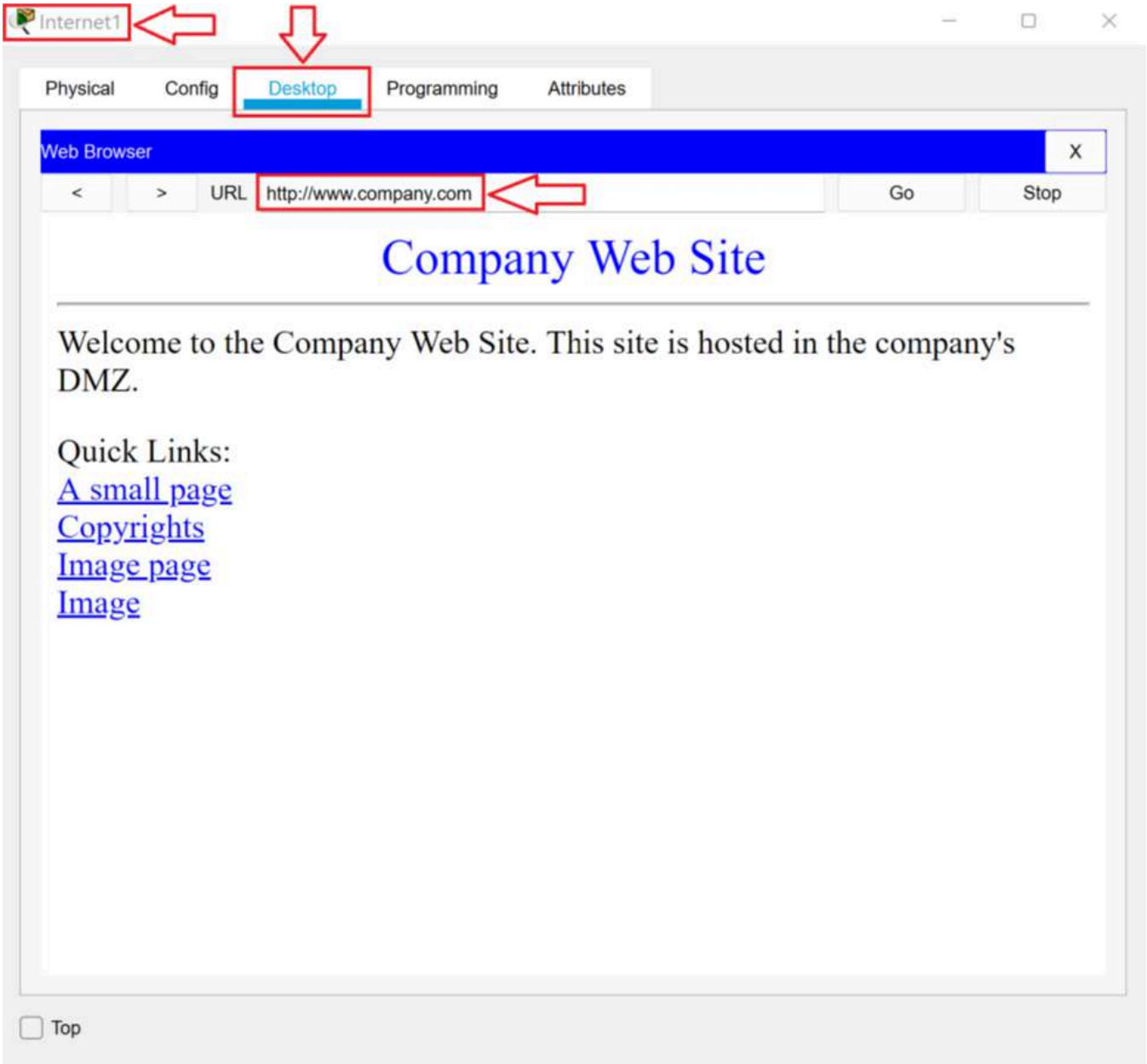
3. Click **Sniffer2** to open the **Sniffer2 Properties** dialog box. On the **GUI** tab, select the **HTML** packet. Notice that the same addresses are displayed. Click the **Clear** button and then close the **Sniffer2 Properties** dialog box. NAT is working on Router1 but you will enable it on Router2 to increase security.



TASK F

In this task, you will test Internet connectivity to the company web server.

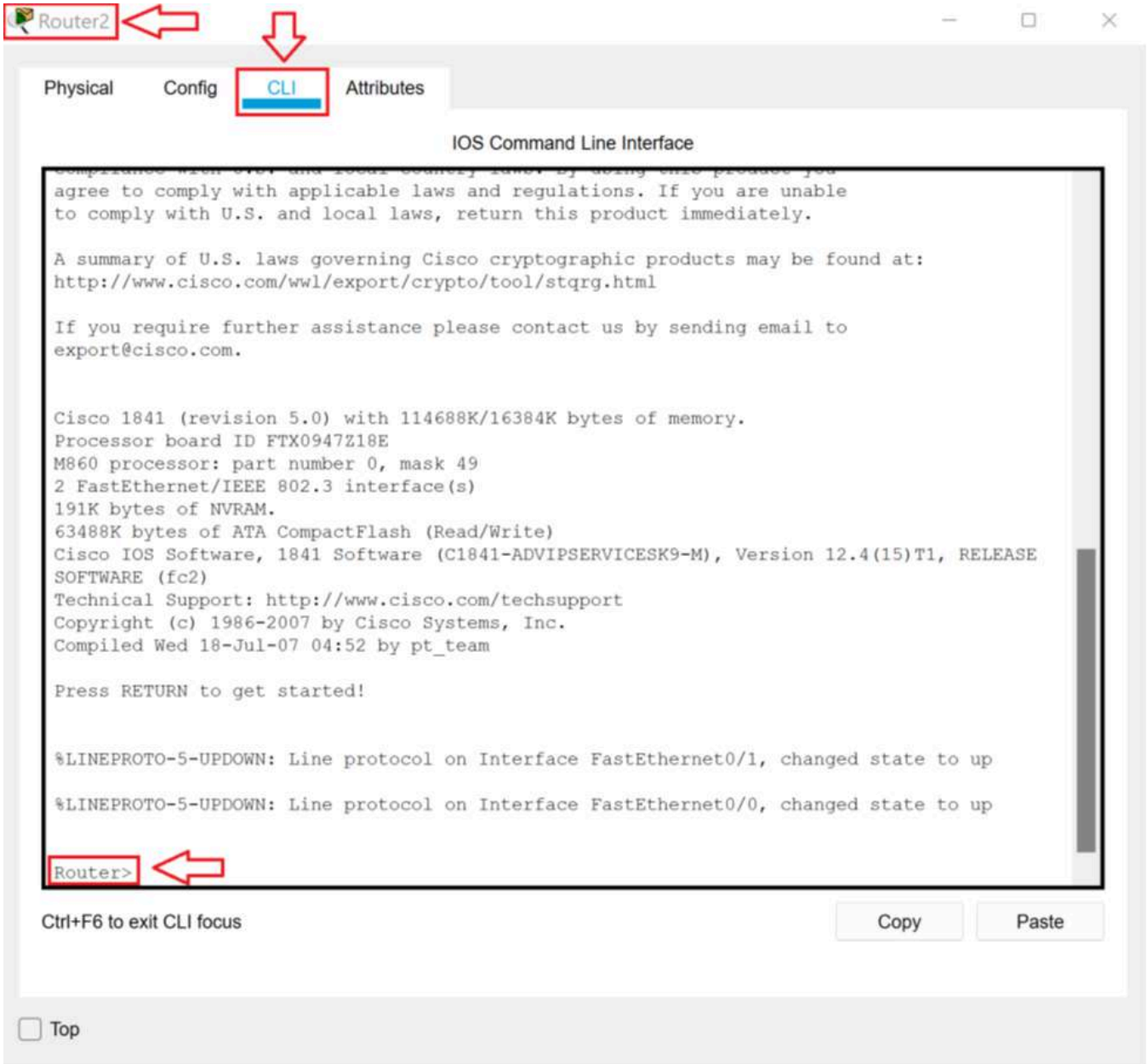
1. Click **Internet1** to open the **Internet1 Properties** dialog box. On the **Desktop** tab, in the **Web Browser**, in the **URL** text box, type **www.company.com** and press **Enter**. Notice that the Internet client can still access the company web site.



TASK G

In this task, you will configure Router2 for NAT.

1. Click **Router2** to open the **Router2 Properties** dialog box. On the **CLI** tab, click inside the box and then press **Enter**.



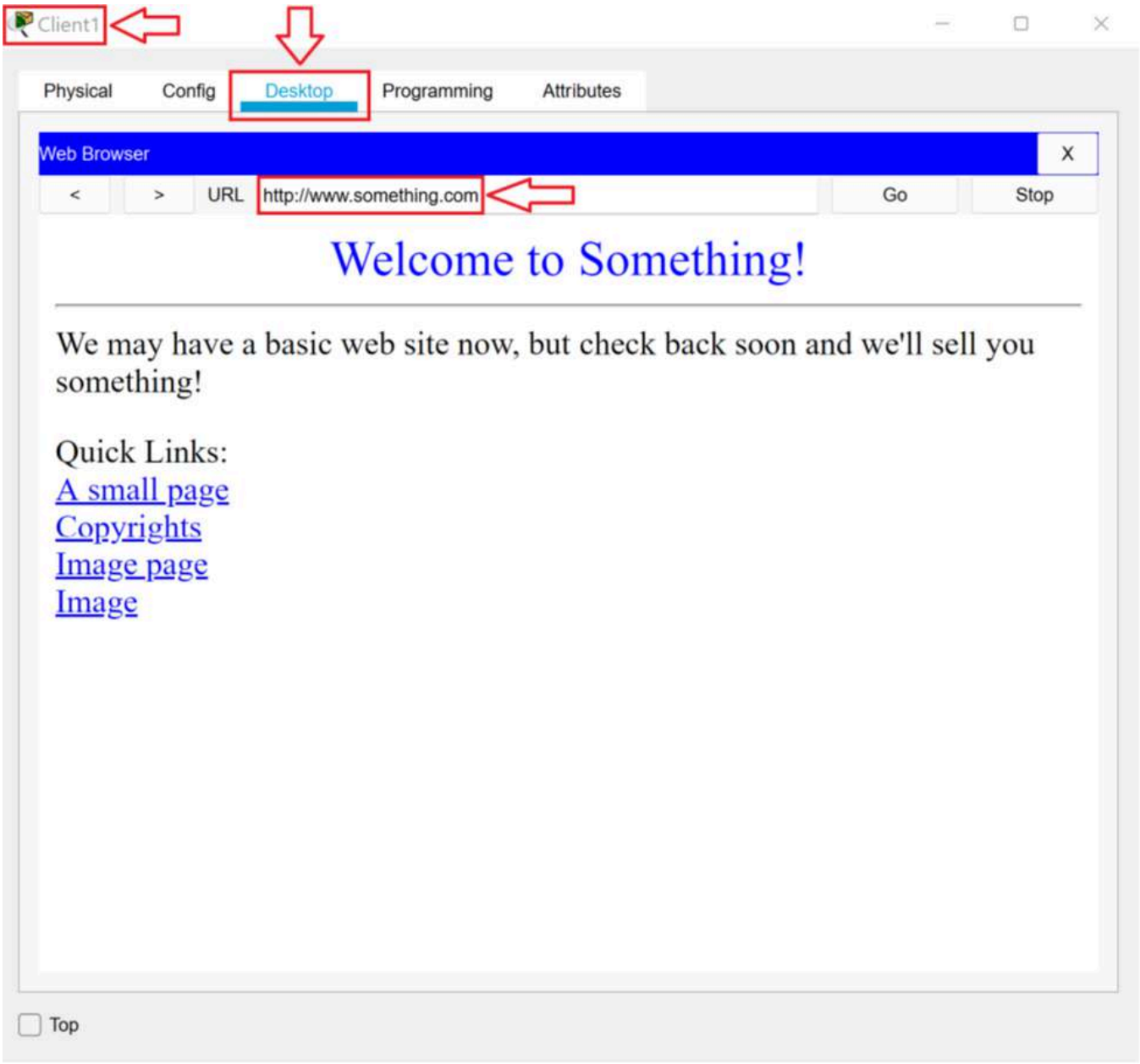
2. Type the following commands, pressing **Enter** after each line. Don't forget to add a space between the network and node addresses in the access-list command.

```
Router>
Router>enable
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int Fa0/0
Router(config-if)#ip nat inside
Router(config-if)#int Fa0/1
Router(config-if)#ip nat outside
Router(config-if)#access-list 1 permit 192.168.2.0 0.0.0.255
Router(config)#ip nat inside source list 1 interface Fa0/1 overload
Router(config)#
```

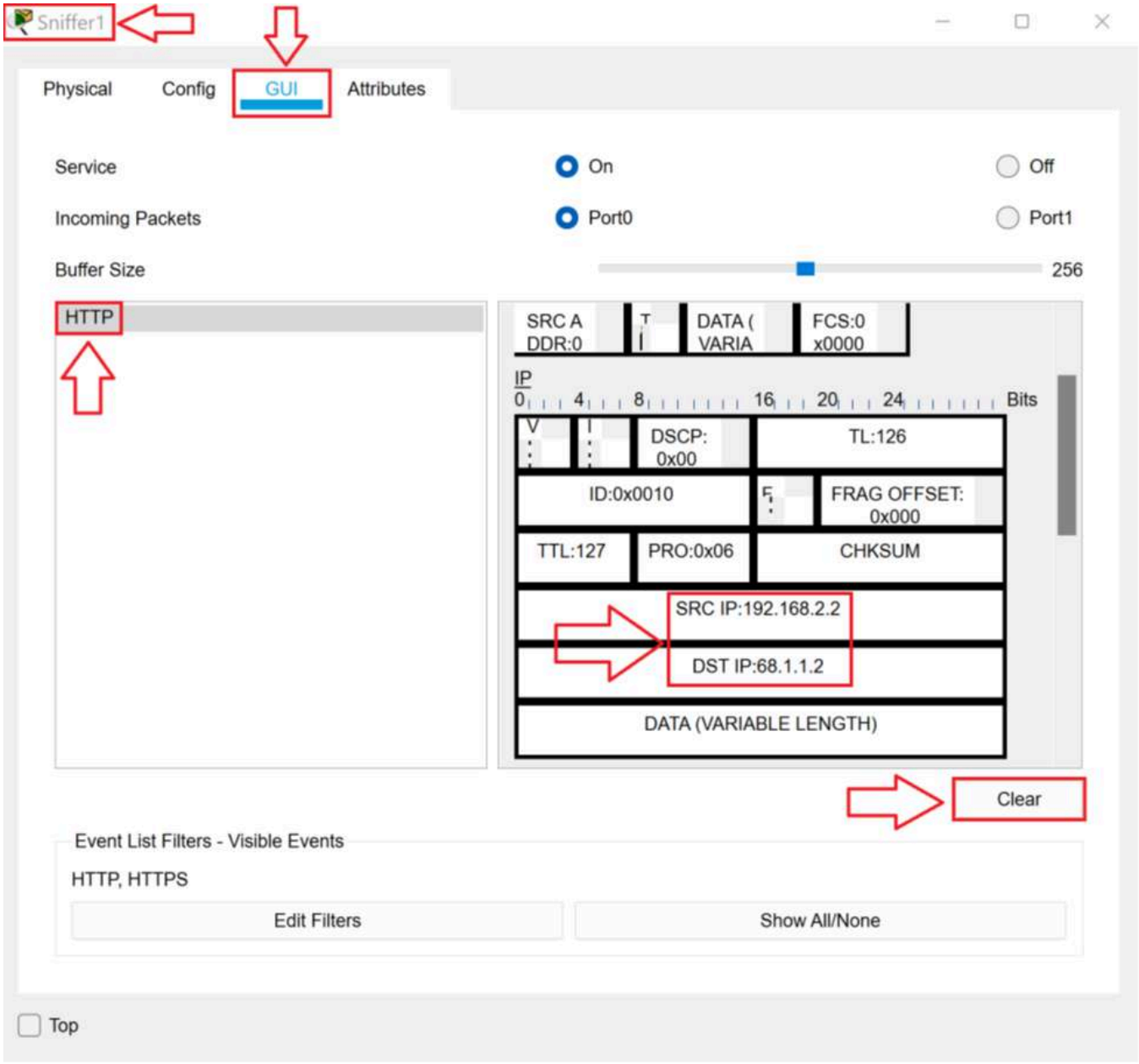
TASK H

In this final task, you will test the full solution.

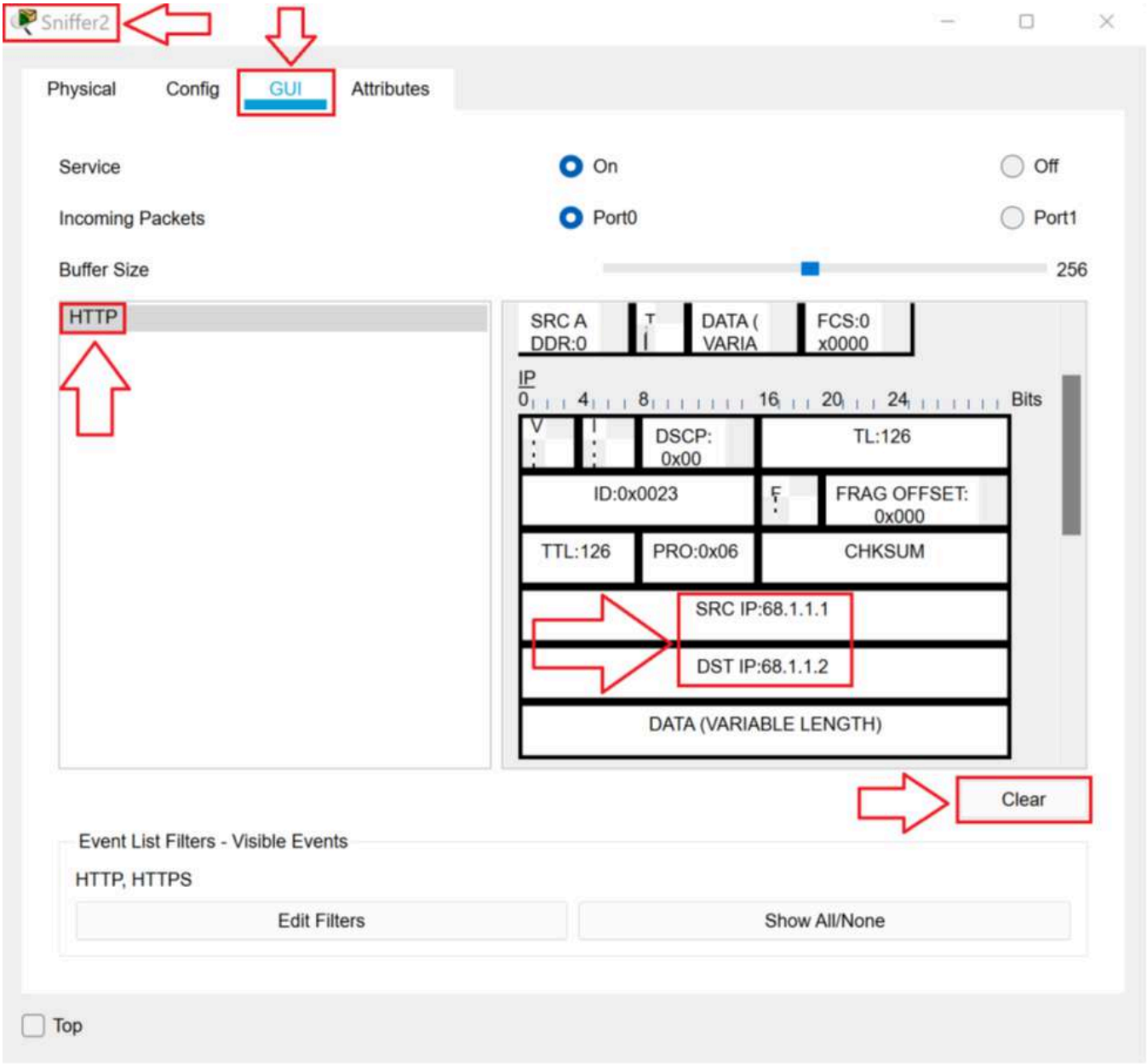
1. Click **Client1** to open the **Client1 Properties** dialog box. On the **Desktop** tab, in the **WebBrowser**, in the **URL** text box, type **www.something.com** and press **Enter**. Click **Clear** and then close the **Client1 Properties** dialog box.



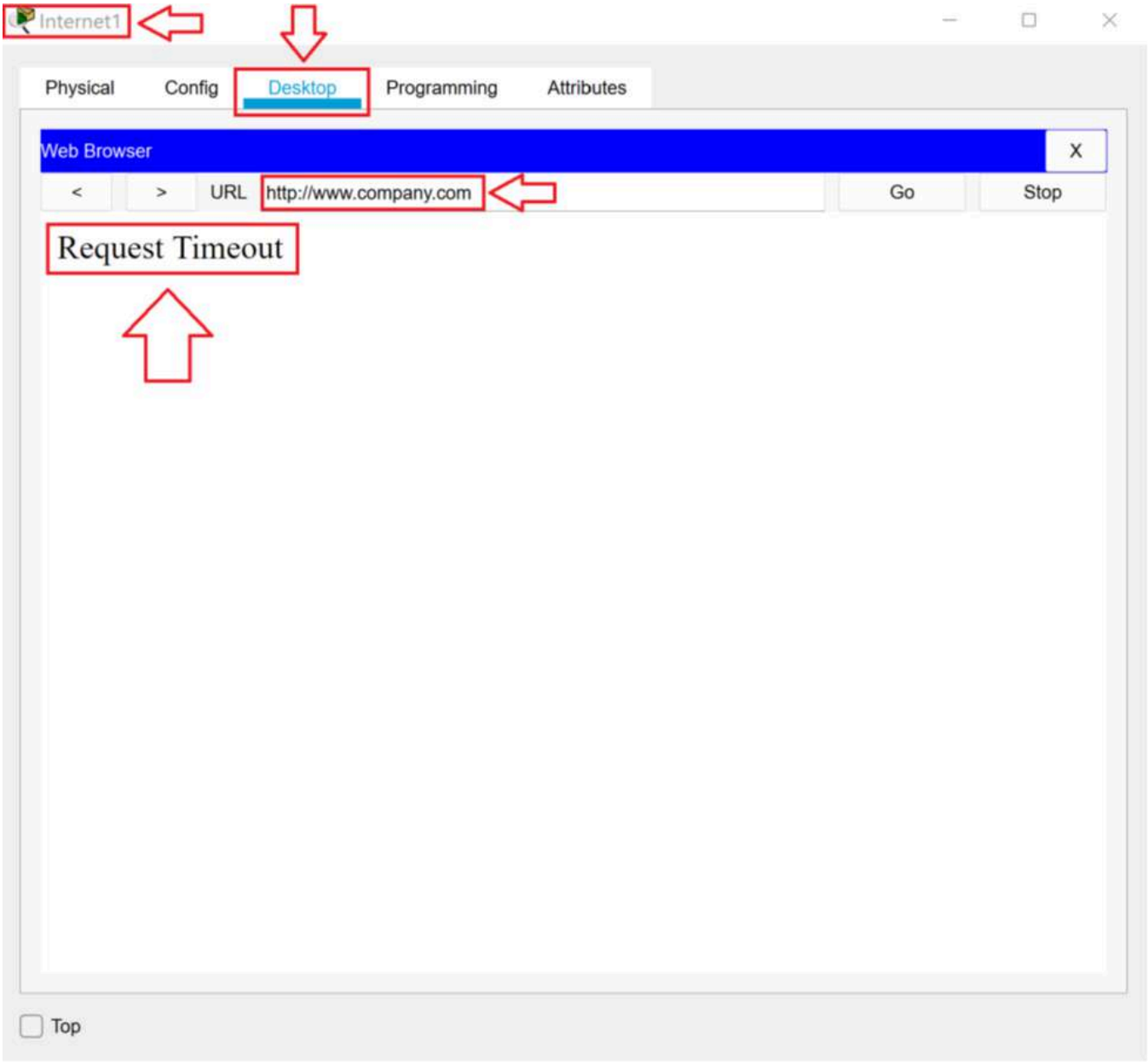
2. Click **Sniffer1** to open the **Sniffer1 Properties** dialog box. On the **GUI** tab, select the **HTML** packet. Notice that the source is the source IP address of Router1 and the destination address is the address of www.something.com. Click the **Clear** button and then close the **Sniffer1 Properties** dialog box.



3. Click **Sniffer2** to open the **Sniffer2 Properties** dialog box. On the **GUI** tab, select the **HTML** packet. Notice the source is Router2 and the destination is www.something.com. Click the **Clear** button and then close the **Sniffer2 Properties** dialog box.

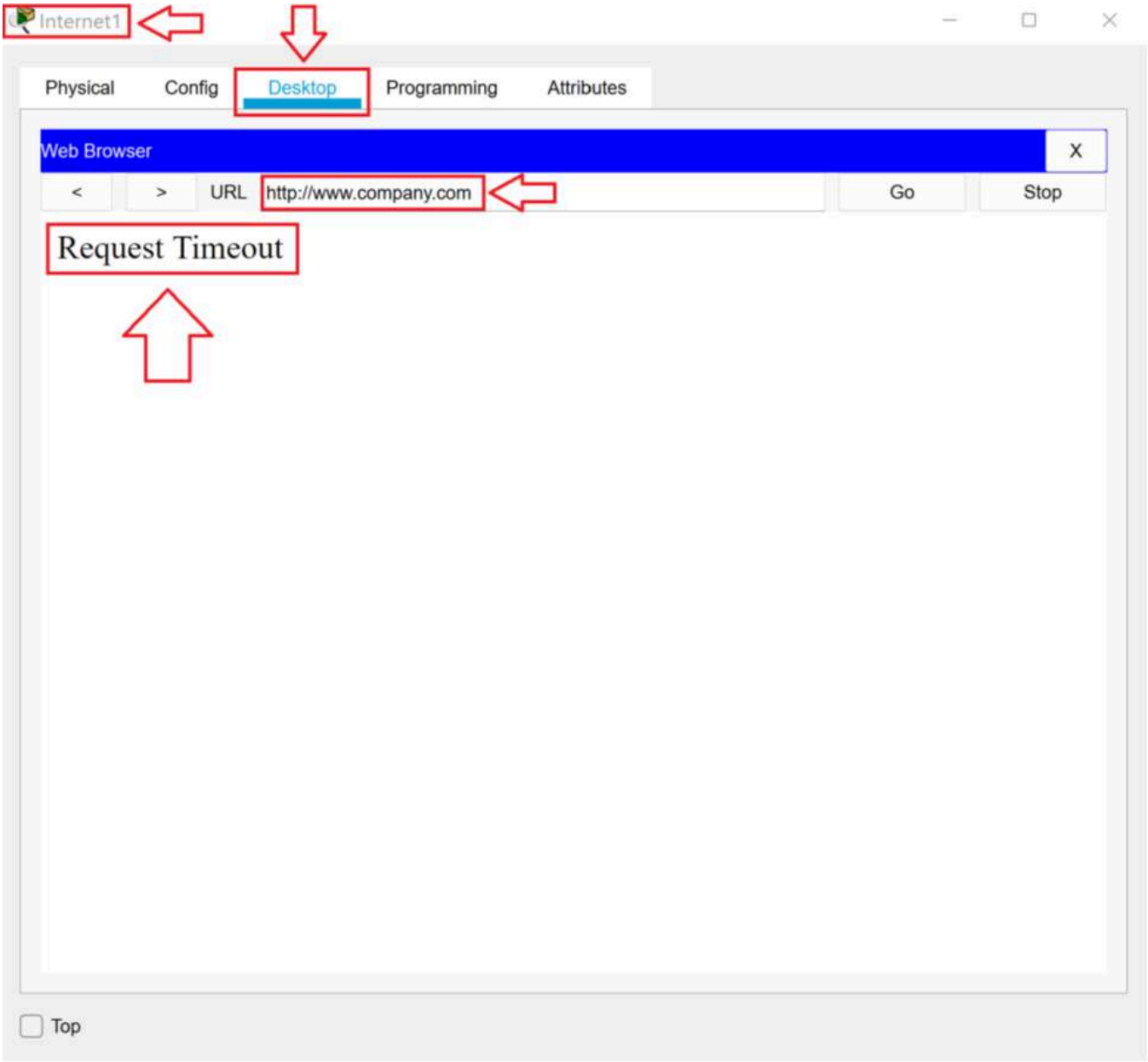


4. Click **Internet1** to open the **Internet1 Properties** dialog box. On the **Desktop** tab, in the **Web Browser**, in the **URL** text box, type **www.company.com** and press **Enter**. Notice that the Internet client now is not able to access the company web site. Now that NAT is enabled on Router2, it discards all traffic that is not a reply. To fix the problem, we need to enable port forwarding on Router2. Close the **Internet1 Properties** dialog box.

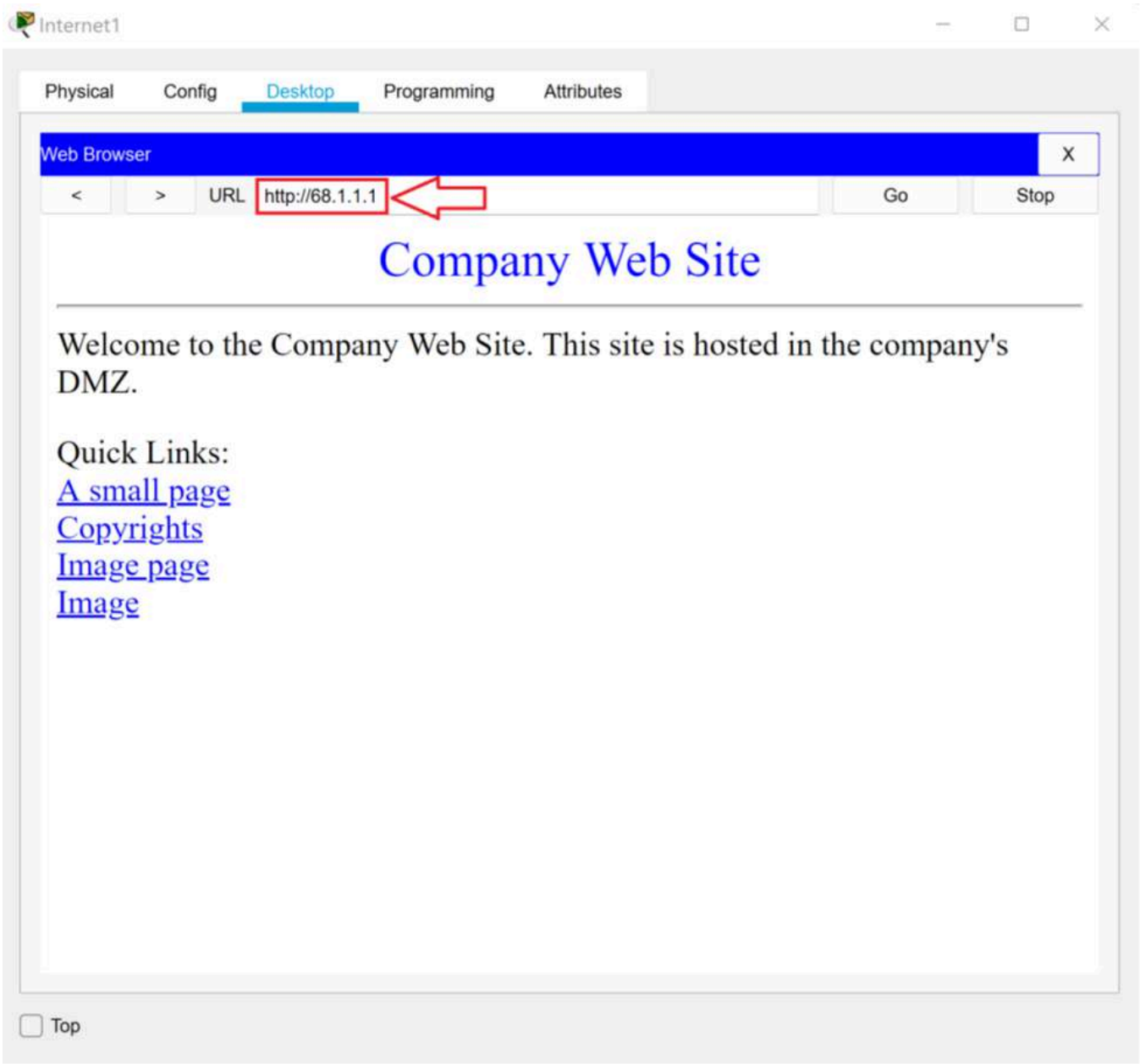


5. Click **Router2** to open the **Router2 Properties** dialog box. On the **CLI** tab, type **enable** and press **Enter**. Type **config terminal** and press **Enter**. Type **ipnat inside source static tcp 192.168.2.3 80 68.1.1.1 80** and press **Enter**. This allows web traffic originating from the Router2 external interface to be passed to the web server in the DMZ. Close the **Router2Properties** dialog box.

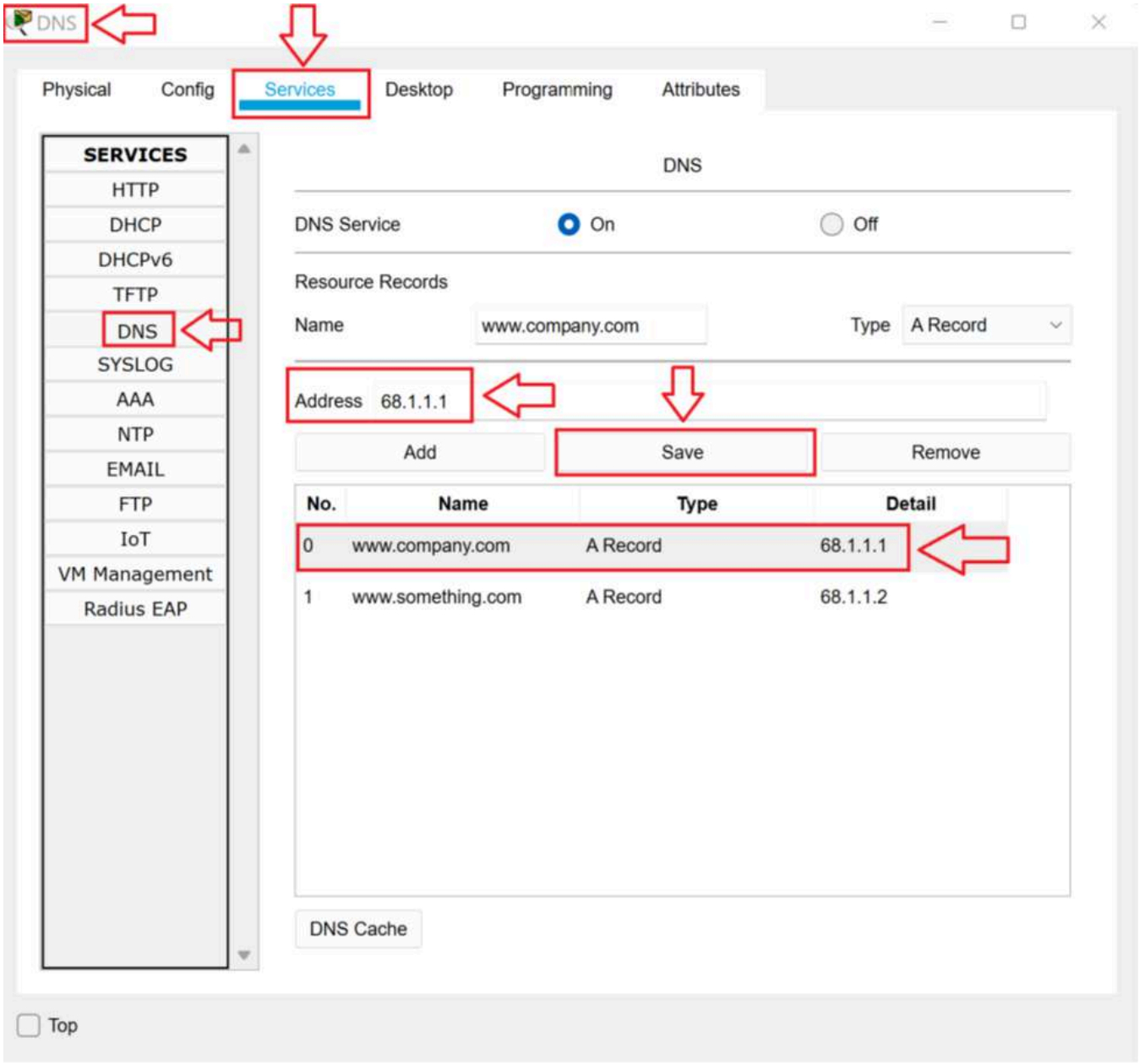
6. Click **Internet1** to open the **Internet1 Properties** dialog box. On the **Desktop** tab, in the **Web Browser**, in the **URL** text box, type **www.company.com** and press **Enter**.Notice that the Internet client still is **not** able to access the web site.



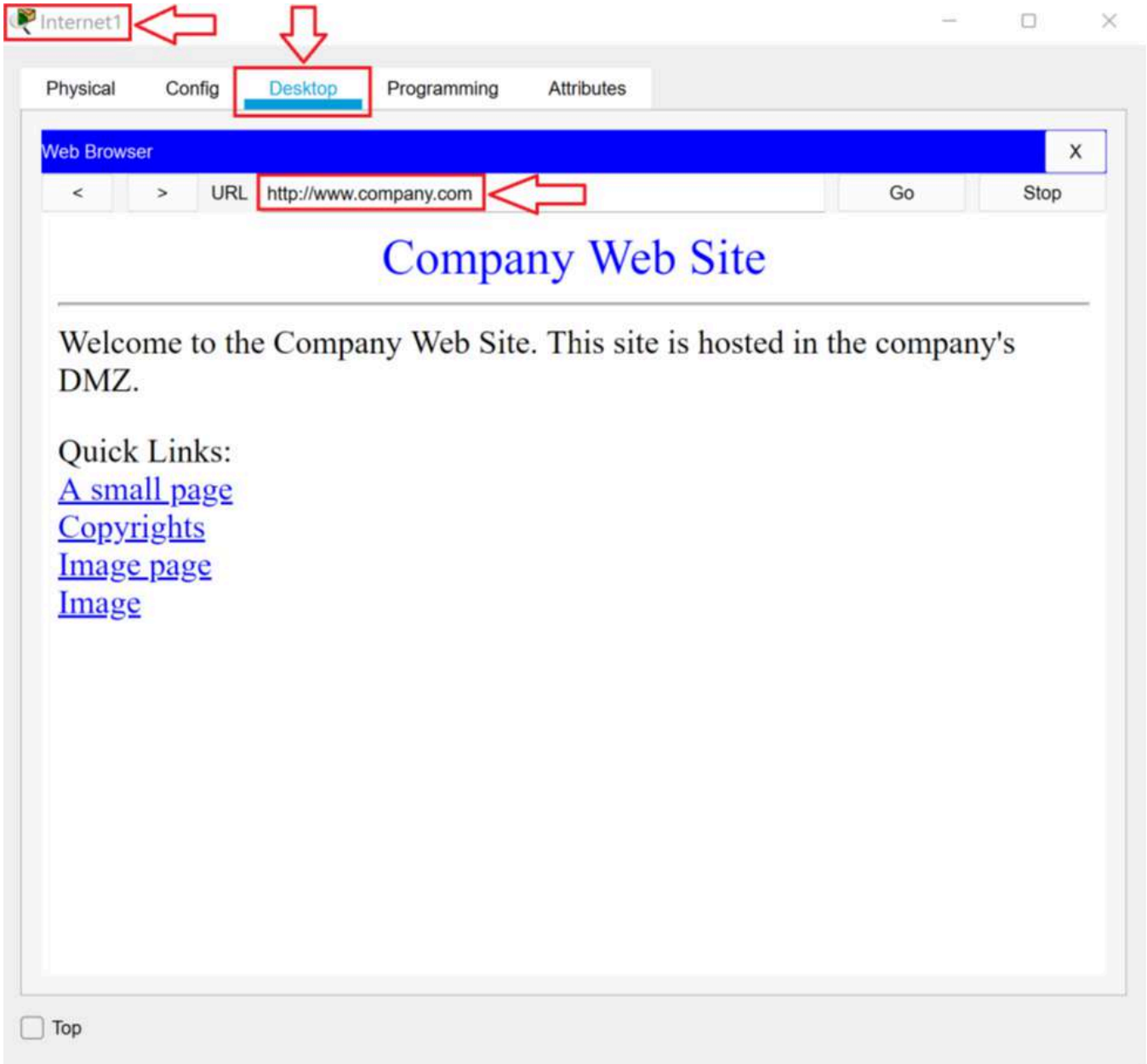
7. In the **URL** text box, type **http://68.1.1.1** and press **Enter**.Notice that the client can access the web site. The external DNS server needs to be adjusted. The DNS server is still resolving the domain name to the IP address of the web server on the DMZ. Now that NAT is creating an obstacle, DNS needs to resolve the address to the external address of Router2. Close the **Internet1 Properties** dialog box.



8. Click **DNS** to open the **DNS Properties** dialog box. On the **Services** tab, in the menu, click **DNS**. Select the **www.company.com** record. Change the **Address** to **68.1.1.1** and then click **Save**. Close the **DNS Properties** dialog box.



9. Click **Internet1** to open the **Internet1 Properties** dialog box. On the **Desktop** tab, in the **Web Browser**, in the **URL** text box, type **www.company.com** and press **Enter**. The site should load properly.



10. Close the **6.3.1 Lab File** file. You do not need to save the changes.

Remote Networking

Remote Access

Remote access means that the user is outside the work network, but they can access to it. Effectively, this might be a user who is on a trip or working from home. Since the Covid19 pandemic, the number of remote workers has exploded.
In remote access networking, the user connects to the work network using a remote connection. Once connected, the user can access resources and function as if it is on the same physical network. The server that supplies remote access also provides security and authenticates users to access the network. All network traffic to and from the remote node passes through the server.
With remote access, the user's experience of the work network will be the same as if they were at work except a little slower.

Remote Desktop

A remote desktop is a connection that enables users to access any network system from their workstation. Once connected, they can perform tasks on the remote system as if they were working at that computer. Remote desktop control uses a special software package to control a remote desktop host computer. Once connected, the remote client can send keyboard and mouse inputs and receive the resultant information on-screen.
Remote desktop is often used to perform administration tasks without having to visit the computer. It can also enable help desk personnel to supply remote assistance.
The difference between remote administration and remote assistance is small. Usually, when you're performing remote administration, if someone was sitting at the host, they could not see what you're doing on their screen. With remote assistance, they can see what you're doing. Some programs also include a chat function.
In some cases, remote desktop hosts may be used as centralized computing. Suppose a few users require access to software that requires expensive hardware to run. They don't need to use this software all the time. It would be expensive to buy each of these users a client computer with that hardware. Also, the company would have to buy a license for the software for each user. Instead, they could install one copy on a remote desktop host. The remote desktop host would be a server that has the expensive hardware. The users could use remote desktop to access the software when they need it.
Remote desktop, as it's used here, is a generic term. However, Microsoft also has a software named Remote Desktop that performs this function. There is other software from many vendors that can perform this service.Some examples are Symantec pcAnywhere®, GoToMyPC®, LogMeIn®, and WebEx PCNow®. These applications are primarily used for remote administration.
Microsoft Remote Desktop Services (previously Terminal Services) and Citrix ICA are two products that specialize in supplying remote desktop as centralized computing.

Remote Access vs Remote Desktop

In remote access, the software and processes are run on the user's computer. The user has access to the work network. With remote desktop, the user's computer is just running a software that sends the keyboard and mouse commands across the network and returns the video signal from the host computer. The remote desktop host (the computer being remote controlled) is the only computer running applications and processes.

Remote Access Infrastructure

Information Technology Infrastructure

The term “IT infrastructure” refers to all the components necessary for IT. Infrastructure includes all hardware, software, networks, and facilities needed to deliver and support technology.
The remote access infrastructure is all the components necessary to deliver and support remote access.

Remote Access Service Servers

Remote Access Services (RAS) servers are available from many sources. This is a generic term for any server that supplies remote access to the network. In the 1990s, the most common types of remote access servers were dial-in Servers. A dial-in server had banks of modems.
Modems allow digital computer signals to be sent over analogue telephone lines. Now, because cell phones are so popular, analogue telephones are often called “land lines.” A user can plug a telephone wire from a wired phone jack in their home into the modem connected to the computer. Then, the computer dials the phone number of the dial-in server. The modem at the dial-in server answers the phone and the connection is made.
Modems are limited to slow speeds. You might see a modem at a company as a backup connection to the Internet in case the primary Internet Service Provider (ISP) goes down, but in the last decade they’ve become quite rare.
Most modern remote access is done through a Virtual Private Networking (VPN) server. VPN servers accept incoming VPN connections. They might be software that run on other servers. Or they may be stand-alone hardware devices that only supply VPN services. In that case, they’re often called VPN concentrators.

Authentication, Authorization and Accounting (AAA)

When users are connecting to the work network via remote access, there are two central concerns.
First, you want to make sure that they’re authorized to connect. That means the users must be authenticated.
When you authenticate someone, it means you verify their identity. The user asserts an identity, usually in the form of a username or email address. Then, they verify that identity by supplying something that only the legitimate user would know or have access to, like a password or a pin.
Second, you want to make sure that the communications between the user and the work network are only accessible to the user and the work network. That means that the communication must be encrypted.
When users connect via remote access, it’s possible to have the RAS server authenticate them. However, if the company already has a database that’s used for authentication (a directory) then it’s more convenient to connect the RAS server to the directory.
For that, we would use a AAA server. Authentication, Authorization and Accounting (AAA) servers supply authentication, authorization, and accounting. The authorization can be done by connecting to the company’s directory. The authorization can be done by creating rules on the AAA server. Accountingrefers to keeping track of who connects and when. This means logging all the connections. AAA servers can supply centralized logging.
The two major AAA protocols are Remote Access Dial-In User Service (RADIUS) and Terminal Access Controller Access Control System (TACACS)

RADIUS

Remote Authentication Dial-In User Service (RADIUS) is a protocol that enables a server to supply standardized, centralized authentication for remote users. The RAS server is configured as the RADIUS client. The RADIUS client will pass all authentication requests to the RADIUS server for verification. User configuration, remote access policies, and logging can be centralized on the RADIUS server. RADIUS is

supported by VPN servers, Ethernet switches requiring authentication, WAPs, as well as other types of network devices. 802.1x, port authentication, connects to RADIUS.
RADIUS is an open protocol (not belonging to a particular company) that is implemented by many vendors. RADIUS uses UDP port 1812 for authentication and UDP port 1813 for accounting.

Diameter

Diameter is an AAA framework that originates from Mobile IP concepts and suppliesmany updates and new ideas that RADIUS does not include. Diameter is not backward compatible (backwards compatible means something works with an older version) with RADIUS, but it does supply an upgrade path. Diameter is a stronger protocol that supplies more advanced features but is not as widespread in its implementation due to the lack of compatible products.

TACACS

Terminal Access Controller Access Control System (TACACS) and TACACS Plus (TACACS+) protocols supply centralized authentication and authorization services for remote users. TACACS includes process-wide encryption for authentication while RADIUS encrypts only passwords. TACACS utilizes TCP rather than UDP and supports multiple protocols. Extensions to the TACACS protocols exist, such as Cisco's TACACS+ and XTACACS.
TACACS+, which is Cisco's proprietary product, uses TCP port 49. It also supports multifactor authentication. TACACS+ is considered more secure and more scalable than RADIUS because it accepts login requests and authenticates the access credentials of the user. TACACS+ is not compatible with TACACS because it uses an advanced version of the TACACS algorithm.

Remote Access Protocols

Remote Desktop Protocols

Remote Desktop Protocol (RDP) the backbone of Microsoft's Remote Desktop system. Its capabilities include data encryption, remote audio and printing, access to local files, and redirection of the host computer's disk drives and peripheral ports. In client versions 6.1 and later, any application that can be accessed via the normal remote desktop can serve as a standalone remote application. The server component, the remote desktop host, is available on most Windows operating systems, and a desktop client is available for most operating systems. The server listens on port TCP 3389.
Virtual Network Computing (VNC) is a platform-independent desktop sharing system. VNC client and server software is available for almost any operating system (and for Java), so a VNC viewer on a Linux system can connect to a VNC server on a Microsoft system and vice versa. VNC is not an inherently secure system but does offer varying levels of password and content encryption, depending on the implementation.
The Citrix Independent Computing Architecture (ICA) is a remote terminal protocol used by Citrix WinFrame and Citrix Presentation Server software as an add-on to Microsoft Terminal Services. ICA enhances and expands on the core thin-client (terminal) functionality found in Terminal Services, and provides client support for additional protocols and services.
The X Window system is a protocol that uses a client-server relationship to provide a GUI and input device management functionality to applications. Current X Window systems are based on the X11 protocol and normally used on UNIX- and Linux-based systems to display local applications. Because X is an open cross-platform protocol and relies on client-server relationships, remote connections are often easy to implement.

Remote Access Protocols

Remote access protocols can provide direct dial-in connections via modems, or they can provide connections via ISPs and the Internet.

Point to Point Protocol

Point-to-Point Protocol (PPP) is a remote networking protocol that works on the Data Link layer of the TCP/IP protocol suite. PPP can dynamically configure and test remote network connections and is often used by clients to connect to networks and the Internet. It also provides encryption for passwords, paving the way for secure authentication of remote users. To log on to a remote session via PPP, you need to enable a remote authentication protocol.
The Point-to-Point Protocol over Ethernet (PPPoE) and Point-to-Point Protocol over ATM (PPPoA) are more recent PPP implementations used by many DSL broadband Internet connections.

Extensible Authentication Protocol

Extensible Authentication Protocol (EAP) is an authentication framework that has many variations. EAP variations are often used for remote access authentication and in 802.1x.

Password Authentication Protocol

Password Authentication Protocol (PAP) is a remote-access authentication method that sends client IDs and passwords as plaintext. Plaintext means that the IDs and passwords are not encrypted. It is generally used when a remote client is connecting to a non-Windows PPP server that does not support password encryption. When the server receives a client ID and password, it compares them to its local list of credentials. If a match is found, the server accepts the credentials and allows the remote client to access resources. If no match is found, the connection is terminated.

Challenge Handshake Authentication Protocol

Challenge Handshake Authentication Protocol (CHAP) is a RAS protocol that uses an encryption method to transmit authentication information. CHAP uses a challenge-response mechanism and authenticates without sending passwords as plaintext over the network.The server sends a “challenge” which is a randomly generated code. The client encrypts the challenge using the user’s password and sends the result back to the server. The server decrypts the response using the password it has on file. If it correctly decrypts the challenge, the client must have used the correct password and the user is authenticated.

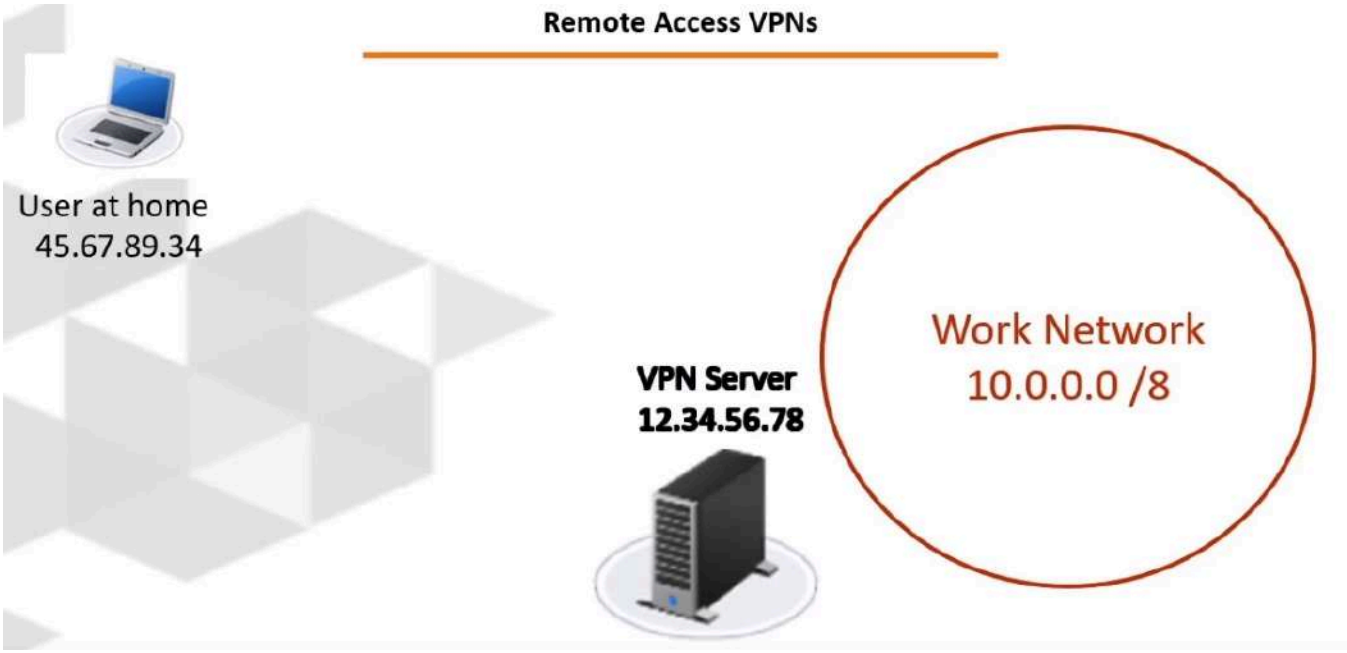
Virtual Private Networking

VPNs

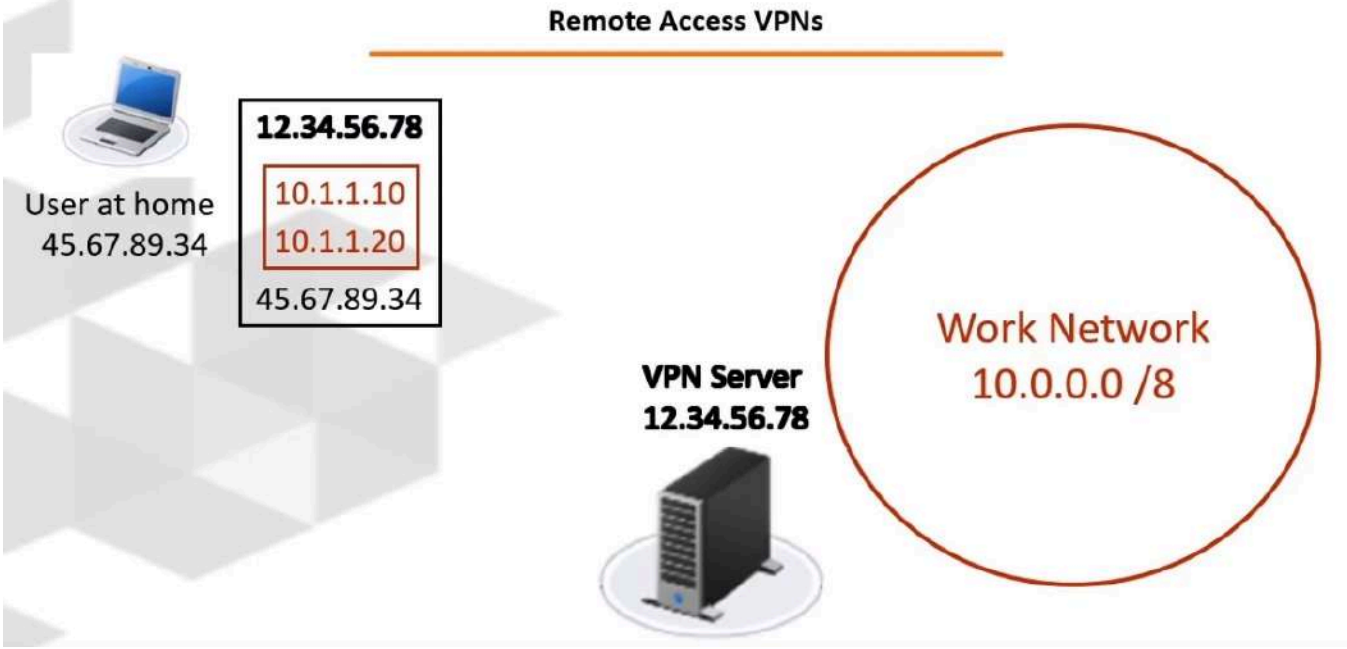
A VPN is a private connection through a public network. And we call it a private connection because the data on VPNs is always encrypted. So it's always private. The public network part is that VPNs typically connect devices over the Internet and the Internet is of course a public network.

There are two types of VPNs, remote access VPNs where the user is remote from the work network and they get access, and site-to-site VPN that connect two sites.

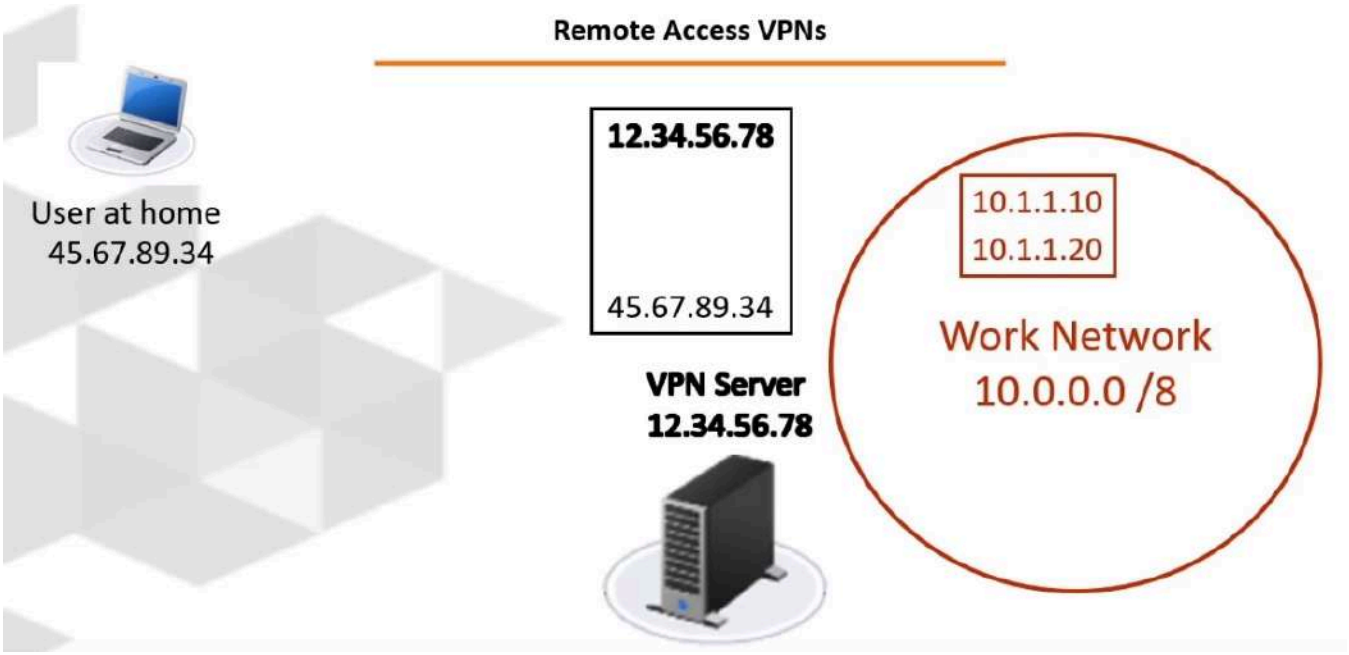
Let's take a look at remote access VPNs. So here we have a user at home. I just made up an IP address. They've got 45.67.89.34. There's a company VPN server at another made-up IP address 12.34.56.78. And all the addresses on the work network start with 10.



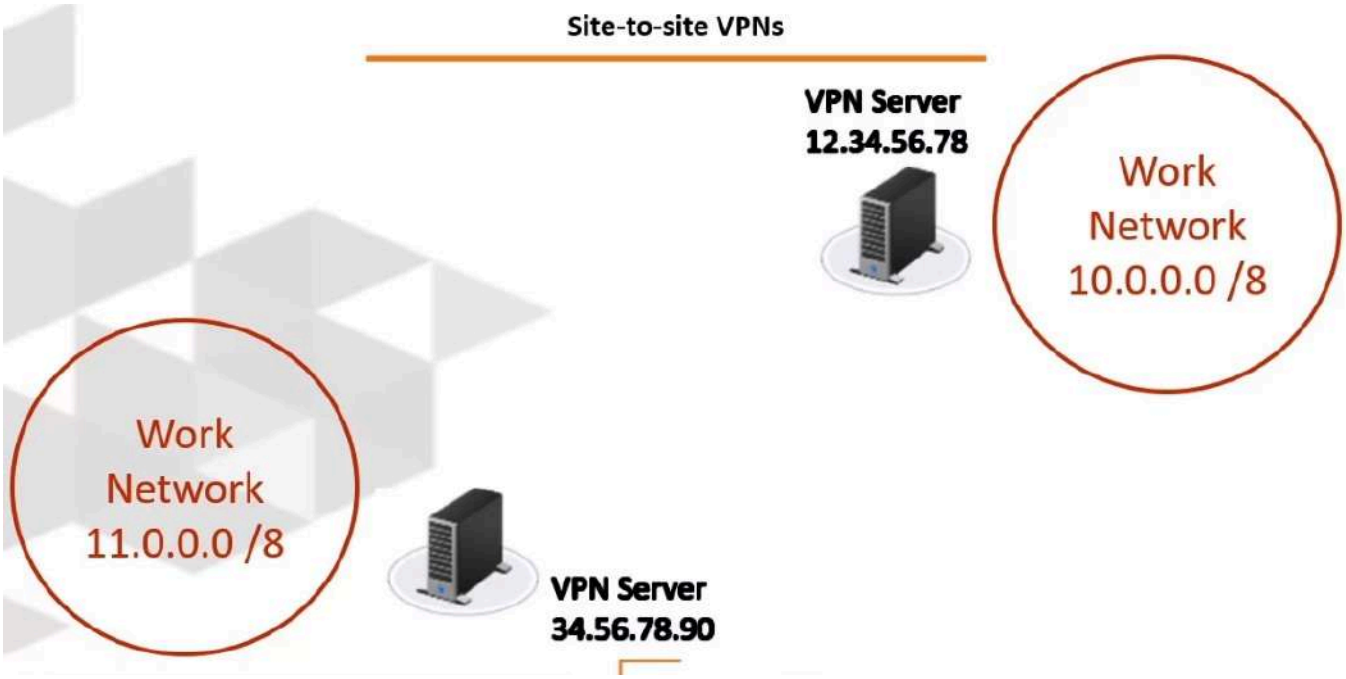
The big thing that a remote access VPN does is when you connect to it, you get an IP address on the work network. At that point, it's like you're connected to the work network if you had plugged in a wire. When traffic needs to be sent to the work network, the client's computer at home makes up a packet from its work network address let's say 10.1.1.10 to whatever work network address it's talking to, let's say 10.1.1.20. And then it encrypts that inner packet. And that idea of a packet within another packet is called tunneling. Now that encrypted packet gets put inside of another packet from the user's address to the VPN server's address.



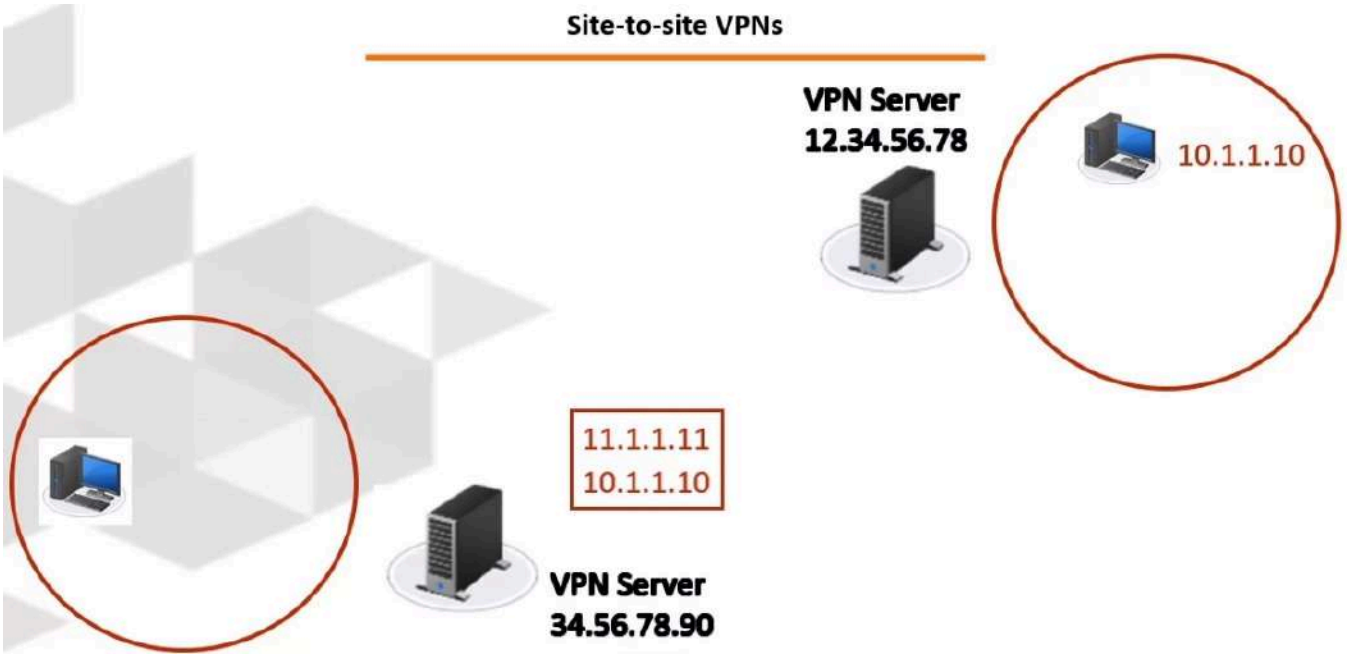
And that's going to travel through the Internet. The VPN service job is to decrypt the inter-packet and put it onto the work network.



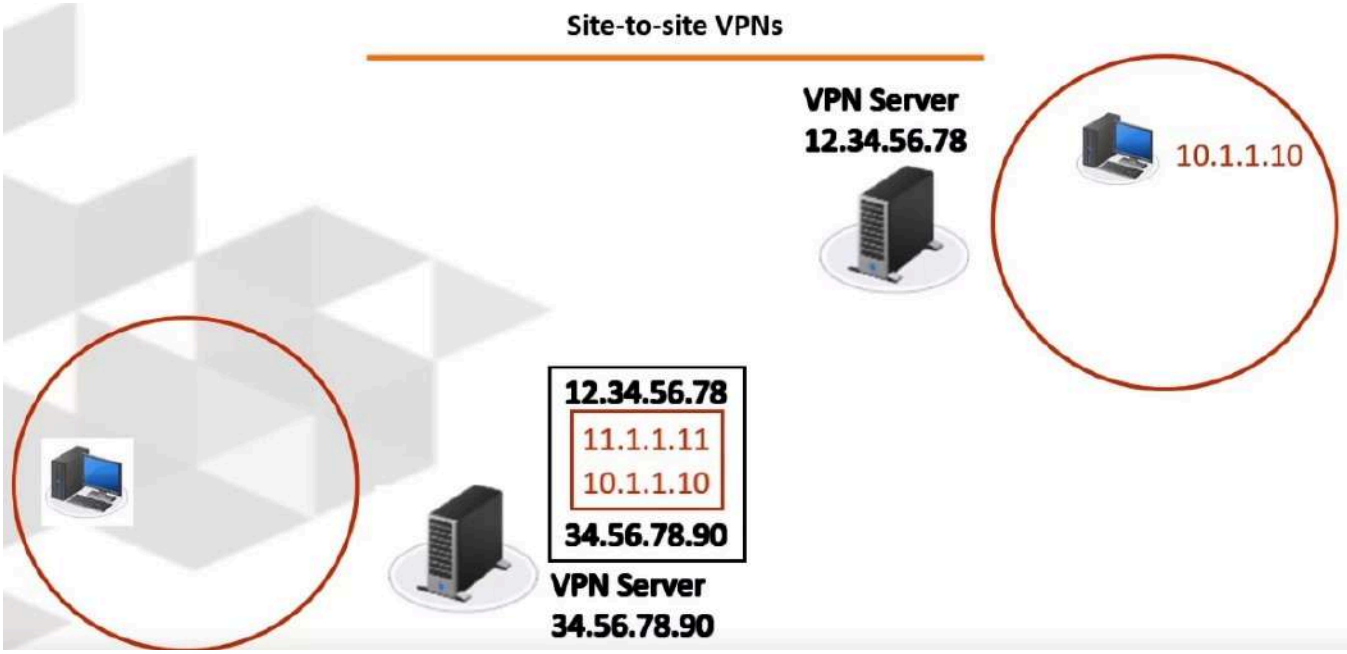
Site-to-site VPNs work a little bit differently. So with the site-to-site VPN, there are two company sites and we want to provide connections between them. So over here in the bottom left-hand corner we have a work network that all the addresses start with 11. And then in the upper right corner, we have a work network where all the addresses start with 10. And there's a VPN server at each site.



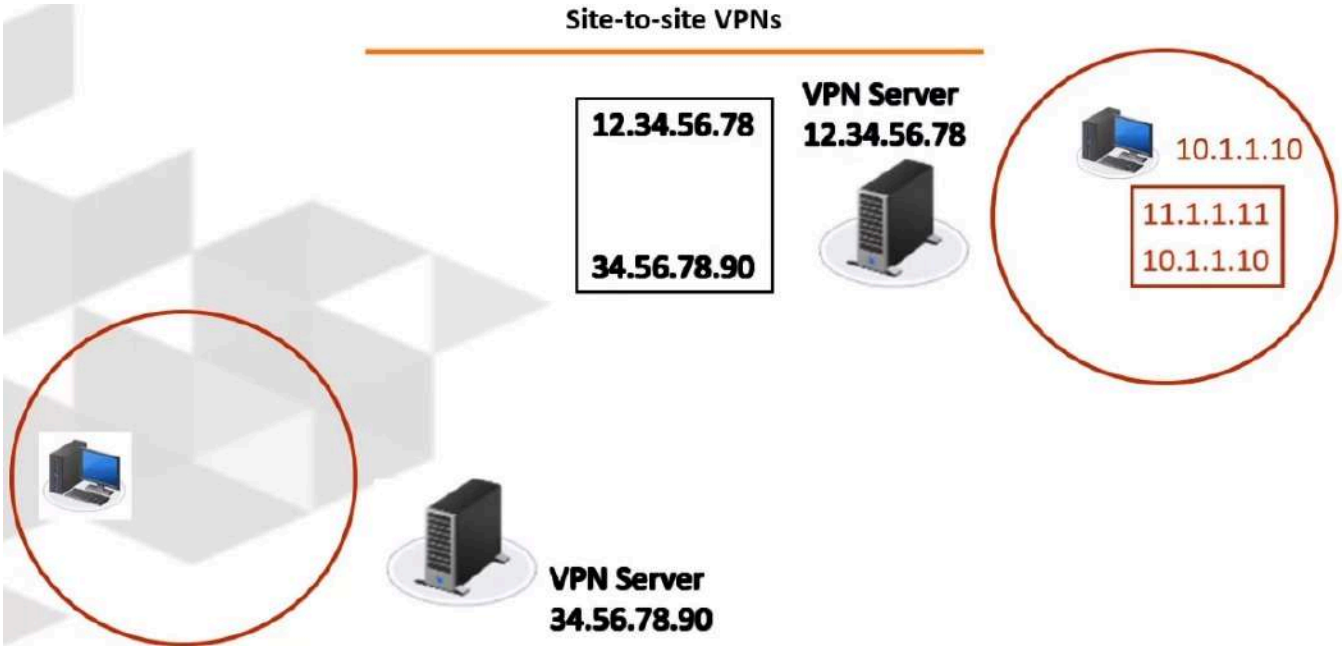
When one of the computers, let's say one down here in the 11 network wants to communicate with something in the 10 network, it makes up a packet from its 11 address to the other computers 10 address. And that gets sent to the VPN server.



The VPN server encrypts that packet and puts it inside of an outer packet from its public IP address to the public IP address of the other VPN server that travels encrypted through the Internet.

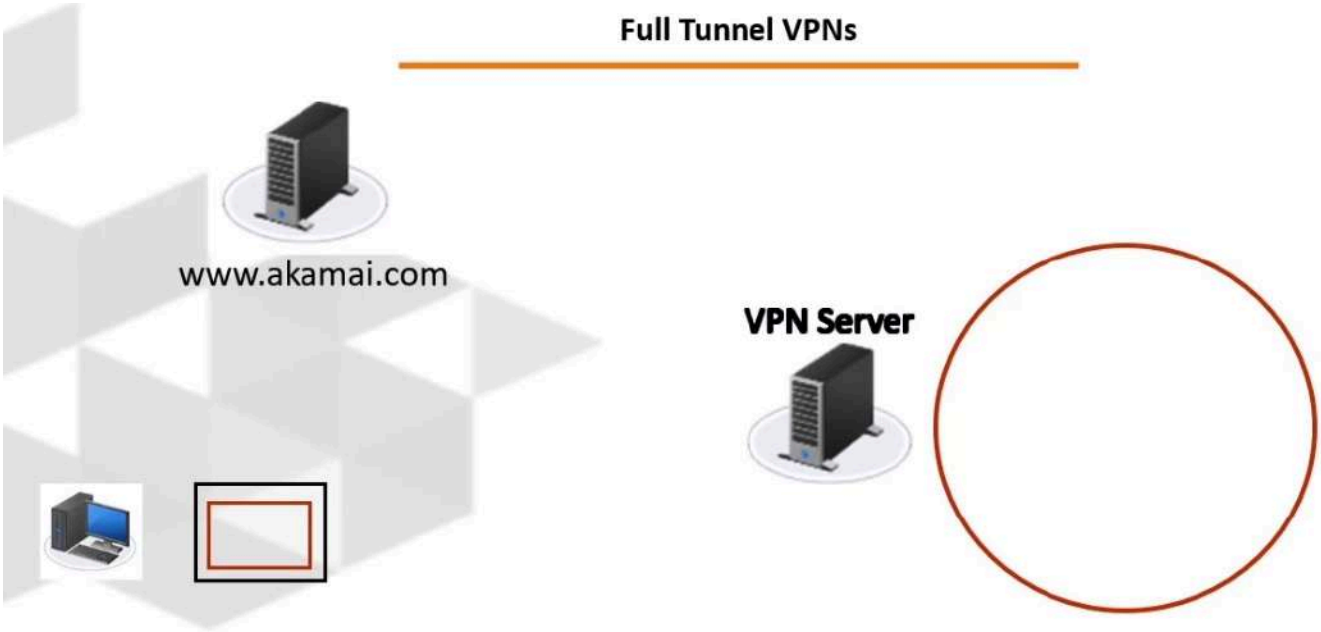


And then the other VPN server's job is to extract the inter-packet and put it on the other work network. So the packets are not encrypted in each of the work networks, it's just encrypted when it enters the tunnel between the two sites and that's a site-to-site VPN.

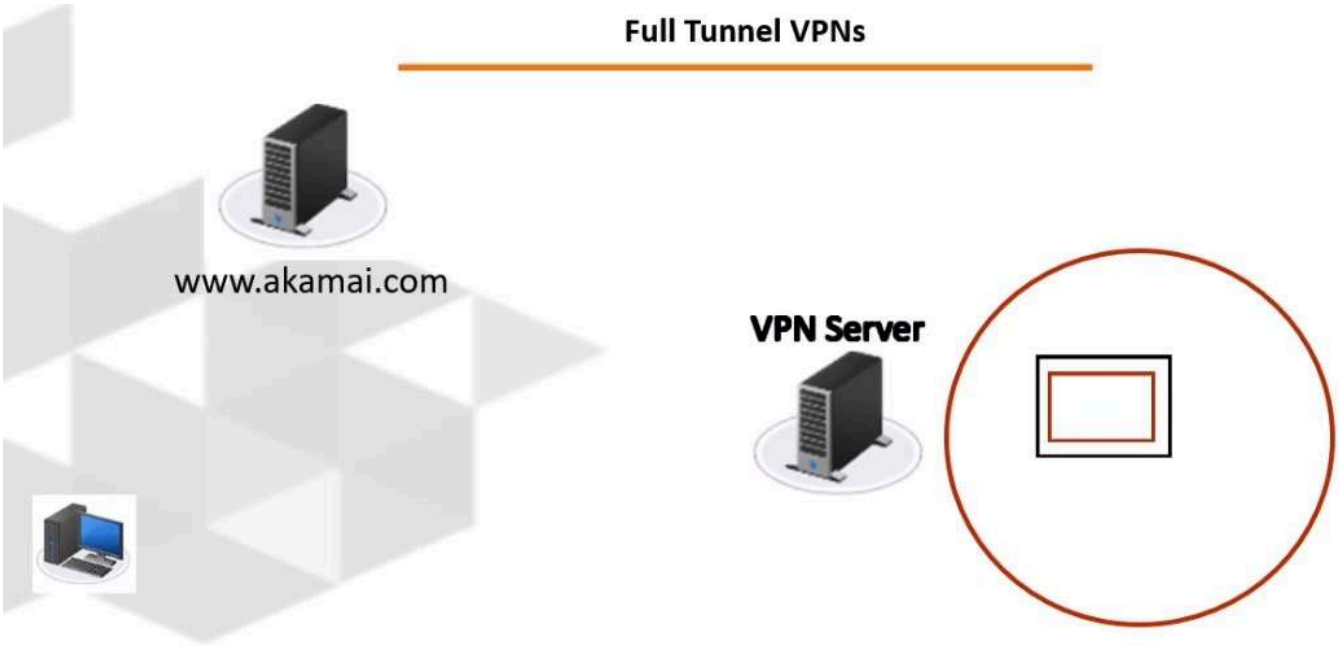


When the VPN client is configured, it can be configured as either full tunnel or split tunnel.

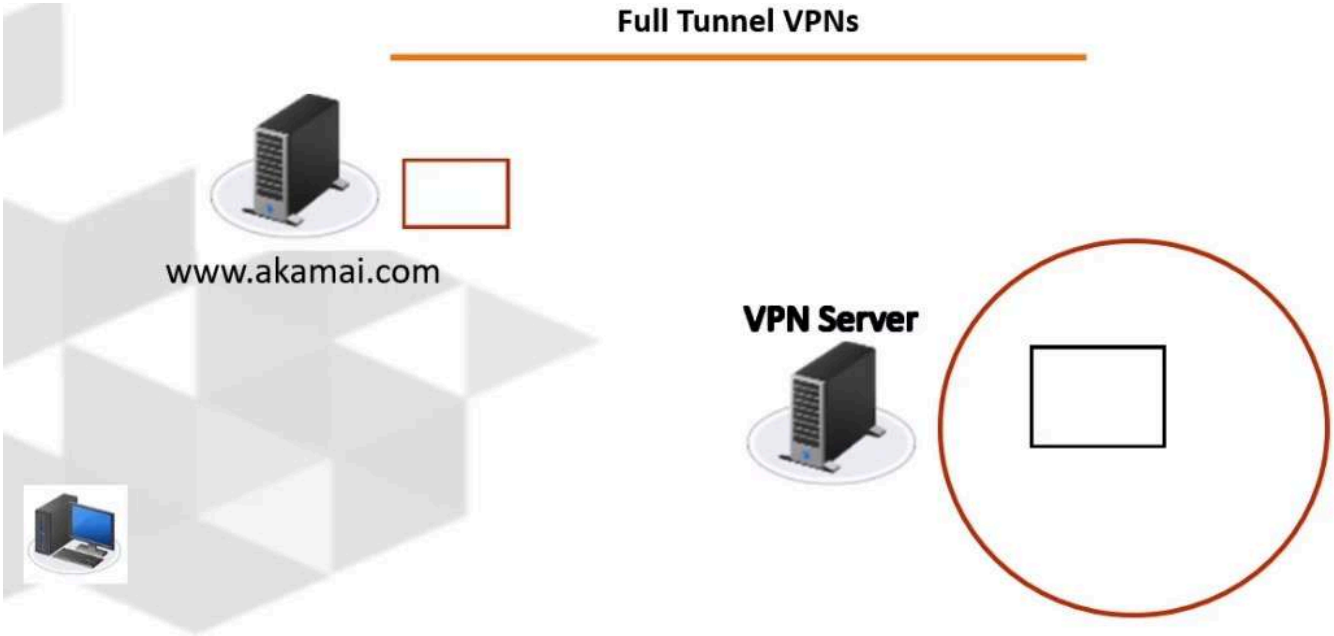
In a full tunnel VPN, the client sends all the data through the VPN, not just the data intended for the work network. For example, suppose this user at home over here on the left is going to access a site on the Internet, www.akamai.com. The user is connected using a full tunnel. It makes up the request to akamai.com



and then that gets sent through the VPN server to the work network.

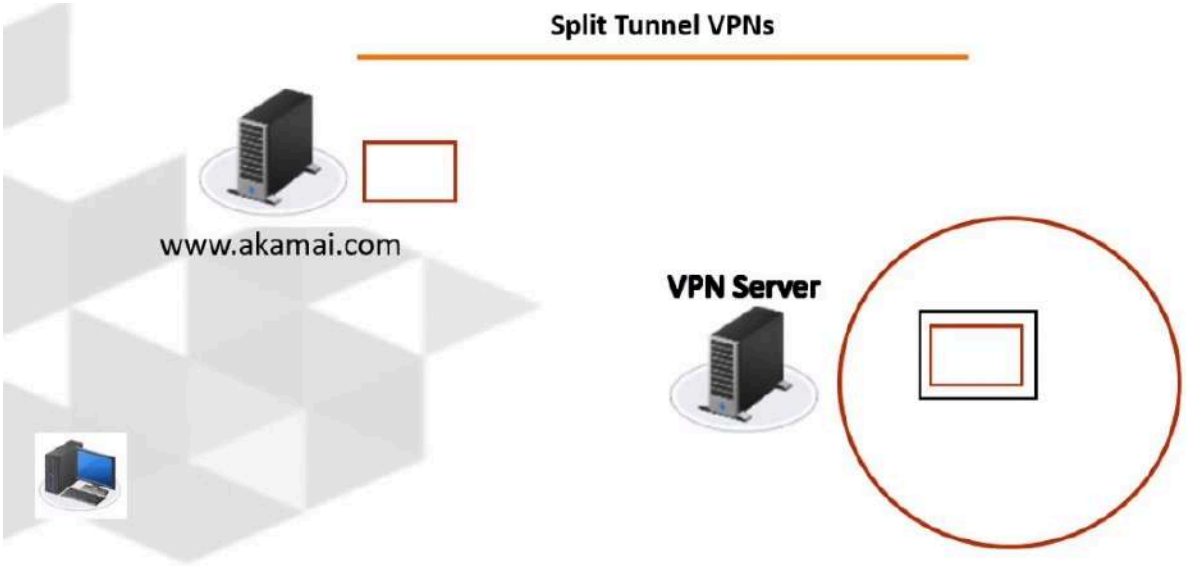


The VPN server extracts it and sends it to the Internet site.



And then when Akamai replies, it'll go back exactly the same way, all to go back through the VPN server and then back to the user. The good thing about a full tunnel is the company has a full record of requirements traffic. It can enforce all the company's regulations about web traffic exactly as if the computer were physically on the work network. The bad thing about this is that it adds significant delayed the process. The VPN client is already connected to the Internet. Sometimes it's such a significant delay that the web page times out and can't be accessed. If that's an issue and there's no security need for a full tunnel VPN, the client can use a split tunnel.

In a split tunnel VPN, data intended for receivers on the work network is sent through the VPN. Data intended for receivers on the Internet are sent directly to the receivers on the Internet. This is significantly faster than routing that traffic through the VPN. However, it could be considered a security risk because the company can't track or regulate that activity.



We talked about VPNs, virtual private networks, the private connection through a public network. It's private because the inner packet that's tunneled inside of an outer packet is always encrypted. We looked at the two main types of VPNs, remote access VPN where the user is remote to the work network and gets access by picking up an address on the work network and sending data to the work network encrypted. And site-to-site VPNs where data is encrypted between two sites that each have a VPN server. We also talked about full tunnel VPNs where all the traffic is sent through the work VPN and split tunnel where internet traffic is sent directly to the Internet and only the work traffic is sent through the VPN.

VPN Protocols

Point to Point Tunneling Protocol

Point-to-Point Tunneling Protocol (PPTP) is a Layer 2 Microsoft VPN protocol that increases the security of PPP by providing tunneling and data encryption for PPP packets. It uses the same authentication methods as PPP. PPTP is the most widely supported VPN protocol among older Windows clients. Deployed over public, unsecured networks such as the Internet, PPTP encapsulates and transports multi-protocol data traffic over IP networks. PPTP is very easy to set up. However, it uses Microsoft Point to Point Encryption (MPPE) which is not very robust encryption. PPTP might be used by smaller companies who have Microsoft servers and can’t afford a better solution. Or if the clients are older Microsoft operating systems that can’t support a more secure alternative. However, it should not be a preferred VPN solution. Microsoft has since developed more robust VPN protocols.
PPTP uses TCP port 1723.

Layer 2 Tunnelling Protocol

Layer Two Tunneling Protocol (L2TP) combines the capabilities of PPTP and Layer 2 Forwarding (L2F) (an older Cisco VPN protocol that did not provide encryption) to enable the tunneling of PPP sessions across a variety of network protocols. L2TP was specifically designed to provide tunneling and security

interoperability for remote access and site-to-site VPNS. L2TP does not provide any encryption on its own. L2TP packets appear as IP packets because, like IP packets, they also have a header, footer, and error correction. As a result, L2TP usesIPSec as the transport for authentication, integrity, and confidentiality. For many years, L2TP/IPSec has been the standard for VPNs. LT2P uses UDP port 1701.

IPSec

IPSec is a secure (encrypted) network protocol suite. IPSec supports mutual authentication (where both sides verify the other's identity). It's one of the most secure encryption protocols available. IPSec can be combined with other VPN protocols. In that case, it handles the encryption. Or, it can be used by itself as a VPN solution.

Secure Sockets Layer (SSL) VPN

Secure Sockets Layer (SSL) was created to secure web pages. Since then, its encryption has been used to secure other protocols. SSL has flaws. Transport Layer Security (TLS) is a replacement for SSL. TLS is not backwards compatible with SSL.

"SSL VPNs" can use SSL or TLS for encryption but regardless of which protocol is used, they're all called SSL VPNs. The advantage of an SSL VPN is that it uses the same port as HTTPS; TCP port 443. This port is rarely if ever blocked by firewalls.

VPNs are not only used for remote access. VPNs can be used in any situation where you want to make sure the communication is encrypted. Suppose you connect to a WiFi hotspot. As an open network, it doesn't have any encryption. You can connect to a VPN provider to make sure the traffic is encrypted.

VPNs can also be used to avoid censorship. The most important thing about encryption is that the data can only be read by the designated recipient. On a network that employs inspection of traffic for enforcing censorship, encrypted traffic can't be inspected. In that case, the network administrator must make decision whether to allow or deny the uninspected traffic. For that reason, networks that employ censorship may block ports 1723 and 1701 to prevent users from using VPN to defy policy. Those users can use an SSL VPN and avoid the regulation.

For example, I have a Nook tablet from Barnes and Noble, a bookstore in the United States. While on a business trip to the United Kingdom, I wanted to buy a book, but the sale wouldn't go through. I spoke to Barnes and Noble and they told me they blocked sales from the UK because they didn't have a license to sell books in that country. I used a VPN to connect to a network in the US. Once it looked like the sale was coming from the US, I could buy the book.

That's something to be aware of and keep in mind if you're the administrator setting the rules. But when data is tunneled through a common protocol like HTTPS or even DNS, it's almost impossible to block.

Microsoft has a proprietary version of SSL VPN called Secure Socket Tunneling Protocol (SSTP).

Internet Key Exchange Version 2

Internet Key Exchange Version 2 (IKEv2) is a VPN protocol developed by Microsoft and Cisco. IKEv2 is one of the fastest VPN protocols. It also supports VPN reconnect. If the Internet connection fails, the client is disconnected from the VPN. VPN reconnect allows the VPN to be automatically reconnected when Internet access is restored.

IKEv2 uses IPSec for encryption. It uses UDP port 500.

Network Monitoring

Network Management

Once you have a functional network, you need to manage that network. Network management involves three main components:

1. Preventing problems
2. Detecting and resolving problems
3. Detecting and addressing problems that occurred in the past

Preventing Problems

The key to preventing problems is to have a good network design. It can be difficult to plan a network. No matter how specific the client is about how the network will be used, it's not possible to anticipate what the network will need to do in the future. Users may underestimate their needs. Technology changes and improves. Ten years ago, 4 Gb of RAM was a lot. Now, that's the minimum for any client computer.

The one thing you can assume is that once you implement technology, you should behave as if that technology is critical. Downtime occurs when a service stops working. In an ideal world, there would be no downtime. In the real world, downtime is inevitable. The best you can do is create a plan to minimize it.

The key to minimizing downtime is to build systems that are fault tolerant. Systems are fault tolerant if they can withstand a fault without there being any downtime. Fault is a generic term for anything that can go wrong. A fault could be a power outage, a virus, software crashing or even a user making a silly mistake.

From a design perspective, there are two important keys to fault tolerance: security and redundancy. From a security perspective, once you have your system built you need to limit who can make changes and when. To achieve this, you need to set up good procedures for change management. Change management is just the process by which changes can happen. Good change management involves having a plan for how changes are proposed, evaluated, and implemented.

Security also involves limiting access. Whether you use an access control list (ACL) on a file or a router, or even implement a firewall, it's important that only the authorized parties get access to the network resources. And they should only have the access they need to do their jobs.

Redundancy in technology means having two or more of any critical components.That way, if one fails, the other can keep working or be put into production.For example, if your concern is a power outage, your primary component may be the electrical company. Your redundant or backup component may be a generator. If your concern is a switch, maybe you have two. Both could be "live" or maybe just one is in production and the other is ready to kick in if the primary fails.

If you achieve a high enough level of fault tolerance you can achieve high availability. High availability is a rating that expresses how closely systems approach the goal of supplying availability 100% of the time while maintaining a high-level of system performance. For practical purposes, high availability means there is little to no downtime at all.

Detecting and Resolving Problems

Detecting problems requires monitoring the systems in production.
The type of utility used to monitor depends on the system you're monitoring. For example, if you're monitoring a server, you might use a system performance monitor. A performance monitor is a software tool that monitors the state of services, processes, and resources on a system. Performance monitors track one or more counters, which are individual statistics about the operation of different objects on the system, such as software processes or hardware components. Some objects can have more than one instance; for example, a system can have multiple CPUs.
When a counter value reaches a given threshold, it shows that the object of the counter may be functioning outside acceptable limits. For example, for processors you can look at % Processor Time. This counter tells how much of the processor's resources are in use. The typical threshold for this counter is 80%. If more than 80% of the total processing capacity is in use, the processor is overburdened.
Many operating systems include basic network performance monitor tools, or you can obtain more complex third-party tools, including network monitors that are based on SNMP.
If your concern is network traffic, maybe you will use a packet sniffer to look at the traffic. Maybe you will implement an IDS to detect network intrusions.
Resolving problems will involve doing whatever needs to be done to restore performance to an acceptable level. What needs to be done will depend on the problem.

Detecting and Addressing Problems that Occurred in the Past

Performance monitors, packet sniffers and IDSs all can search for issues in real time. A good network should also be configured for a certain amount of auditing. The results of the auditing will be contained in log files.
A log file is a record of actions and events performed on a system.Almost every system will a log, if not several. Those logs need to be managed. Log files take resources, especially storage, on a system. This needs to be configured in advance. When something is going wrong, there are usually many more entries in the log than usual. If the system runs out of storage and needs to overwrite old log entries, there may not be enough entries to troubleshoot. Some companies implement centralized logging where the events in the log files are sent to a central server. If that's the case, there may be software that analyzes the events and generates an alert when something has gone wrong. If that's not the case, someone needs to be assigned to go through the logs to make sure that nothing has gone wrong in the past without generating an alert.
Many security breaches have only been detected after the fact. Looking at historical records of monitoring software can help find that a problem did in fact occur. Then, even though the problem happened in the past, it can be addressed in the present.

Troubleshooting

Troubleshooting is the recognition, diagnosis, and resolution of problems. Troubleshooting begins with the identification of a problem. It doesn'tend until services have been restored and the problem no longer affects

users. Troubleshooting can take many forms. But all approaches have the same goal: solving a problem efficiently with a minimal interruption of service.
A troubleshooting model is a standardized step-by-step approach to troubleshooting. The model serves as a framework for correcting a problem on a network without introducing further problems or making unnecessary changes. Models can vary in the sequence, number and name of the steps involved, but all models have the same goal: to move in a methodical and repeatable manner through the troubleshooting process.

Troubleshooting Steps

These are the steps we will use in troubleshooting:

1. Identify the problem
    a. Question users: what happened? When? How many users is it affecting? Does it always happen?
    b. Re-create the problem: ask the user to show you the problem. If there's no user, try to recreate the problem yourself.
    c. Identify symptoms: what is this affecting? How does the problem occur?
    d. Determine if anything has changed.
2. Establish a theory of probable cause
    a. What do you think is causing the problem?
    b. Question the obvious.
3. Test the theory to determinecause (if possible.)
    a. Determine next steps to resolve the problem.
4. Establish a plan of action to resolve the problem and identify potential effects
    a. Establish a plan of action before you start making changes.
    b. Detail each step that you will take while attempting to resolve the issue.
5. Implement the solution or escalate as necessary
    a. Implement the plan of action step by step to fix the problem.
    b. Or escalate the issue to the proper personnel
6. Verify full system functionality and if applicable implement preventative measures.
7. Document findings, actions, and outcomes.

For example, let's say a user reports that their internet access is down.

1. Ask the user when their access went down. Is it affecting all users or just them? Is it affecting all web sites or just one web site? In this example, the user says that she lost Internet access just after lunch. No one else is having any problems. She isn't sure if it's all web sites, she says she can't get to www.akamai.com in her browser.
2. You need to establish a theory of probable cause. Question the obvious: is she typing the right address in the browser? Is the Akamai web site down? Your theory is that the Akamai web site is down.
3. To test your theory, you open the browser on your computer and type in the address. The web site comes up. Now you must revisit step two. Your new theory is that her network cable is bad. You can test that by confirming that no other users are affected. That means the problem must be between the user's browser and the switch. You can also look at the light on the network card. If its not lit, the network card can't detect the switch.
4. You decide that you will switch out the network able between the computer and the wall jack. You decide to use the network cable from your own computer because you know it is working. If that resolves the problem, you will give your cable to the user.
5. Now, implement the plan.

6. After you switch out the cable, verify that the user can browse internet sites. In this case, there are no preventative measures to take.
7. Now you need to document what caused the problem and what you did to resolve it. (And you need to get yourself another cable!)

Ticketing Systems

Every company should have a system they use to keep track of problems and resolutions. Perhaps a the real “problem” is only revealed in the pattern of events. For example, suppose one day you turn on your air conditioner and the fuse in the circuit breaker flips. You reset the fuse and decide maybe there are too many electrical devices on that outlet. You plug the air conditioner into a different outlet and think the problem is resolved. Then, a week later, you turn on the air conditioner again. The fuse for the second outlet flips. Again, you reset the fuse and move the air conditioner. After the fuse flips for the third outlet, you wonder if there’s a problem with the air conditioner. You have it serviced by a professional and find out it needs to be repaired.

Sometimes the resolution to one problem creates another problem later. In our example, let’s imagine that you move the air conditioner to the second plug. A week later, your partner uses the microwave and the fuse flips. Maybe the actual problem is that now the circuit with the microwave is overloaded. But if your partner doesn’t know that you moved the air conditioner to that circuit, they may incorrectly believe that there’s a problem with the microwave.

Larger companies usually have ticketing systems. These are electronic databases used to track problems and resolutions. In a typical large company, the process for resolving problems might look something like this:

1. Someone or something detects a problem.
    a. If it’s an automatic alert from a monitoring system like SNMP, the monitoring system creates an electronic “ticket” (record in the ticketing system) for the alert.
    b. If it’s a user, the user calls a Help Desk or uses an online form to create a ticket.
2. If the user calls into a Help Desk, Help Desk personnel (first level support) may try to resolve the problem and close the ticket in the same call. If not, they will put notes in the ticket documenting what they did and escalate (forward) the ticket to second level support.
3. Second level support will contact the user or visit the system (in person or remotely) to try to resolve the problem (close the ticket.) If they cannot, it may be escalated to third level support.
4. Third level support will resolve the problem.

Every person who works on the ticket should make notes in the ticket about what was done to resolve it. Whether the ticket is escalated or closed, it’s important to include notes. Even if the ticket is closed, the problem resolved, the notes may help in the future. You will never be scolded for over-documenting a ticket. There is no such thing.

Troubleshooting Network Connectivity

Physical Issues

Components

To troubleshoot physical issues, you need to either have a diagram of how the network is physically connected or at least be able to imagine the physical route traffic should take.
If the two devices are local, the path will be:

1. Sender's NIC
2. Network cable from the sender's NIC to the wall jack
3. Wire from the wall jack to the sender's port in the patch panel
4. Patch cable from the sender's port in the panel to the switch
5. Switch
6. Patch cable from the switch to the receiver's port in the panel
7. Wire from the receiver's port in the patch panel to the wall jack
8. Network cable from the wall jack to the receiver's NIC.
9. Receiver's NIC.

If the two devices are remote, it's the same physical path except the "receiver" will be the default gateway, the NIC on the router connected to the local network.
Efficient troubleshooting can reduce the amount of time to diagnose the issue.Proceed step by step through the troubleshooting steps until the issue is resolved.

Troubleshooting Steps

1. Is this just affecting one sender/receiver or all of them?
    a. If it's all devices, suspect the switch. Replace the switch.
2. Can the sender/receiver connect to other devices?
    a. This will tell you which device to focus on, eliminating either components 1-4 or 6-9.
3. Start from the NIC on the device.
    a. On the NIC will be a "link light" that is lit when the NIC detects another device on the other end of the cable. Is that light lit?
        i. If no, the problem is probably beyond the NIC.
            1. Switch out the cable between the NIC and the wall jack. If the light comes on, go to step ii.
            2. If not, switch out the cable between the patch panel and the switch.
            3. If not, the problem is probably the network card itself, go to step ii.
        ii. If yes, the problem is inside of the computer, or the cable is performing poorly:
            1. Does the speed/duplex on the NIC match the speed/duplex on the switch?
            2. Is the cable too long? Try a different cable.
            3. Switch out the NIC.
        iii. If that does not resolve the issue, the problem probably isn't physical.

Logical Issues

Components

Logical issues really mean issues with TCP/IP.
If the two devices are local, the path will be:

1. Sender's IP address
2. Sender's MAC address
3. Receiver's MAC address
4. Receiver's IP address

If the two devices are remote, it's the same path except the "receiver" will be the default gateway, the NIC on the router connected to the local network.
Efficient troubleshooting can reduce the amount of time to diagnose the issue. Proceed step by step through the troubleshooting steps until the issue is resolved.

Troubleshooting Steps

1. At the sending device, use **ipconfig /all** (**ifconfig**) to look at the IP address:
    a. Does the device have an APIPA address? (169.254.0.0/16)
        i. If yes, the problem is with DHCP.
            1. Is the client having physical connection issues to the network? If so, repair them.
            2. Try **ipconfig /release** and **ipconfig /renew**
            3. If not, repair the DHCP server.
2. Use **ping** to test connectivity to a remote device (any Internet web site will work.)
    a. Result: ping request could not find host. The problem is DNS. (You can confirm this by pinging the remote device by IP address.)
        i. If the DNS server responds, escalate the issue to the DNS Administrator for the domain. The client's DNS server is functioning properly.
            1. Use **nslookup / dig** to lookup any name that you know is a good name. Example, **nslookup akamai.com** or **dig akamai.com**
        ii. If the DNS server doesn't respond, use **ipconfig /all** to get the IP address of the DNS server. Ping the server by its IP address.
            1. If the DNS server responds, escalate the issue to the DNS Administrator. Something is wrong with DNS.
            2. If the DNS server does not respond, there is a physical issue.
    b. Result: Destination host unreachable.
        i. Use **ipconfig/ifconfig** to verify that the client has a default gateway.
            1. Is the default gateway on the same network as the client?
                a. If not, there's a problem with the IP address or the default gateway. Set an appropriate combination.
                b. If so, ping the default gateway by IP address.
                    a. Result: Destination host unreachable.
                        a. There is probably a physical problem. Confirm by using the **arp -a** command to verify there is no entry for the default gateway's MAC address in the arp cache.
                    b. Result: Request timed out
                        a. There is no problem with the default gateway. Use the **tracert** command to find where the problem is located and escalate to the appropriate administrator.
        ii. Result: Request timed out
            1. It is likely there is no network connectivity issue. The router knows where to send the packet but is not getting a reply. Most likely ICMP is turned off on the remote device. Try another remote device or try a **tracert** to test this theory.

Wireless Issues

Components

The wireless network is either ad hoc or infrastructure mode.
If the network is ad hoc, the only components are the two wireless NICs on the devices.
If the network is infrastructure (through a central device), the components are the wireless NIC on the client and the WAP.
Efficient troubleshooting can reduce the amount of time to diagnose the issue. Proceed step by step through the troubleshooting steps until the issue is resolved.

Troubleshooting Steps

1. Can the wireless client see the SSID of the network? If not:
    a. Do you need to remove obstacles or get closer to the wireless network? You can increase the transmitting power of the WAP or change the antenna on one of the devices.
    b. Do you have the correct SSID? Has the broadcast of the SSID been disabled?
    c. Is there a standards mismatch? For example, if the WAP only supports 802.11 ac and the client only supports 802.11g.
    d. If it's an ad hoc network, is the device hosting the network advertising the network?
2. If the client can see the SSID, can it associate to the network?
    a. If not:
        i. Is the client configured to use the same security as the WAP. For example, if the WAP is set to WPA2-Enterprise and the client is set to WPA-PSK it will not associate.
        ii. If the WAP uses a pre-shared key (password/passphrase) are you typing the right one?
3. If the client can associate to the network, do you have network connectivity?
    a. If it's an open network, did the captive portal open? Did you agree to the terms?
4. If you have poor network connectivity:
    a. You may be too far from the wireless network. Sometimes the signal strength is enough to associate but not enough to support meaningful traffic. You can increase the transmitting power of the WAP or change the antenna on one of the devices. You can also try changing the placement of the WAP.
    b. Are there too many wireless signals? Use a WiFi analyzer to generate a heat map. Try a different wireless channel or reduce the number of WAPs until the network is not congested.
    c. Maybe there are too many devices. Reduce the number of devices or upgrade to a WiFi standard that supports MU-MIMO.

Troubleshooting

In this lab, you are being provided with a file that has a number of problems.
Download the **8.3.1 Lab File** and open it in **Packet Tracer**.
8.3.1 Lab File
PKT File
The lab is successfully complete when all clients, including the wireless laptop, can ping each other. The clients must also be able to browse the website www.company.com which is being hosted on web1.hosting.com.
The clients should obtain their IP addresses dynamically. The servers should have static addresses.
There are several ways to solve this lab. Any answer that allows you to achieve the desired outcome is the right answer.
There are no specific steps for this lab. If you have completed all the previous labs, you should have enough experience to resolve all the problems. Refer to previous labs if you get stuck.
Several hints have been provided. You're encouraged to work with the hints if you cannot solve the lab.
A sample answer file has been provided for you. You are strongly encouraged to attempt to complete the lab and to work with the hint files before you download the answer file. If you achieved the goals, you do not need to download the answer file.

8.3.1 Lab File Answer File
PKT File