Network Terminology

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

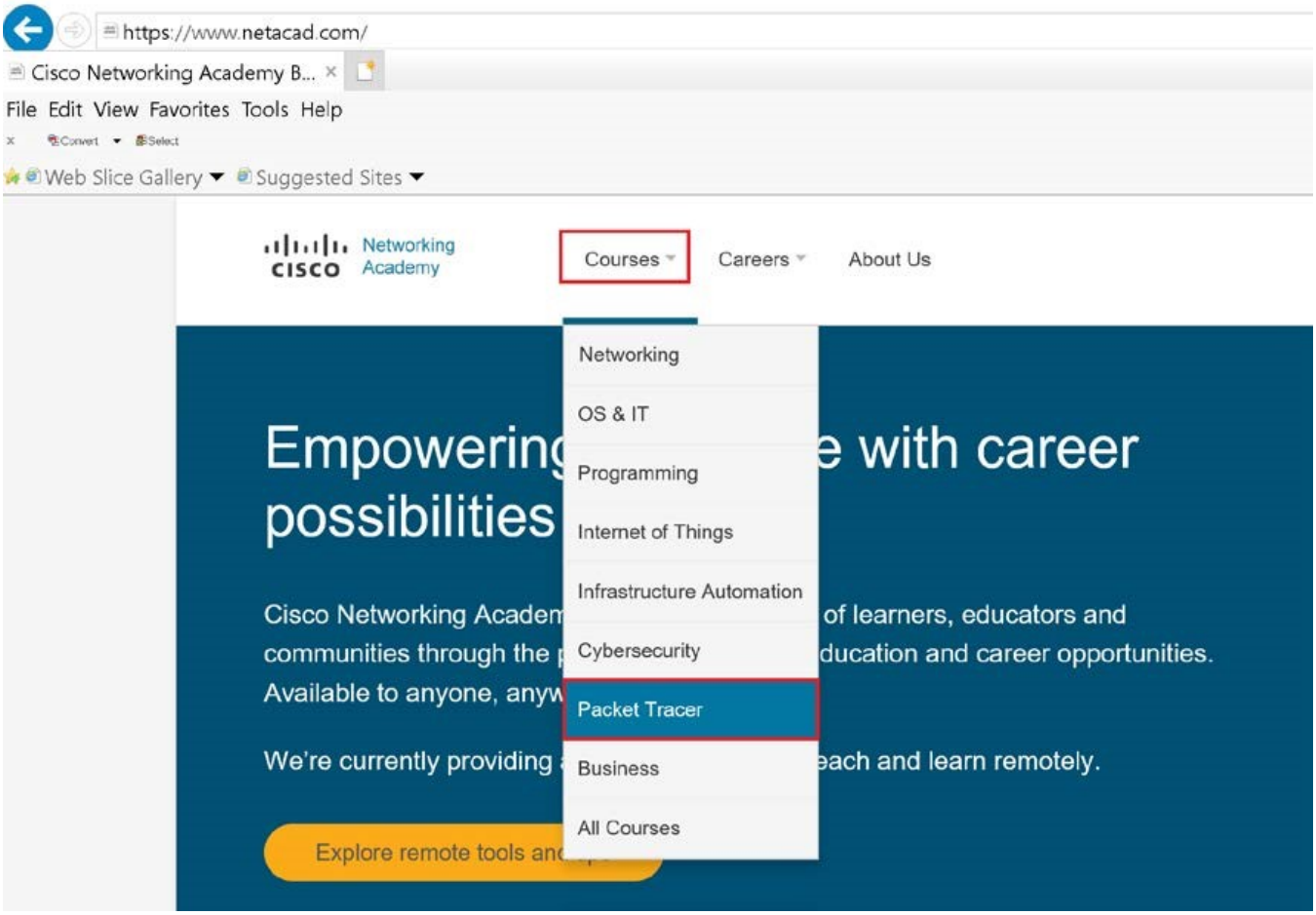
Install Packet Tracer

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

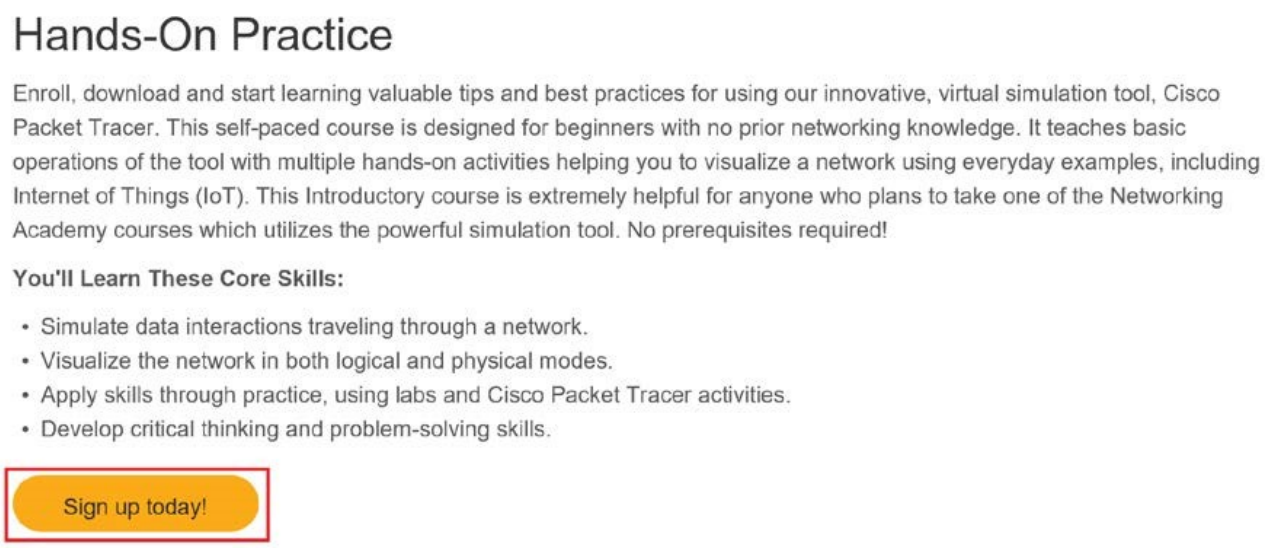
First, you will need to sign up for a course on the Cisco Networking Academy site:

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

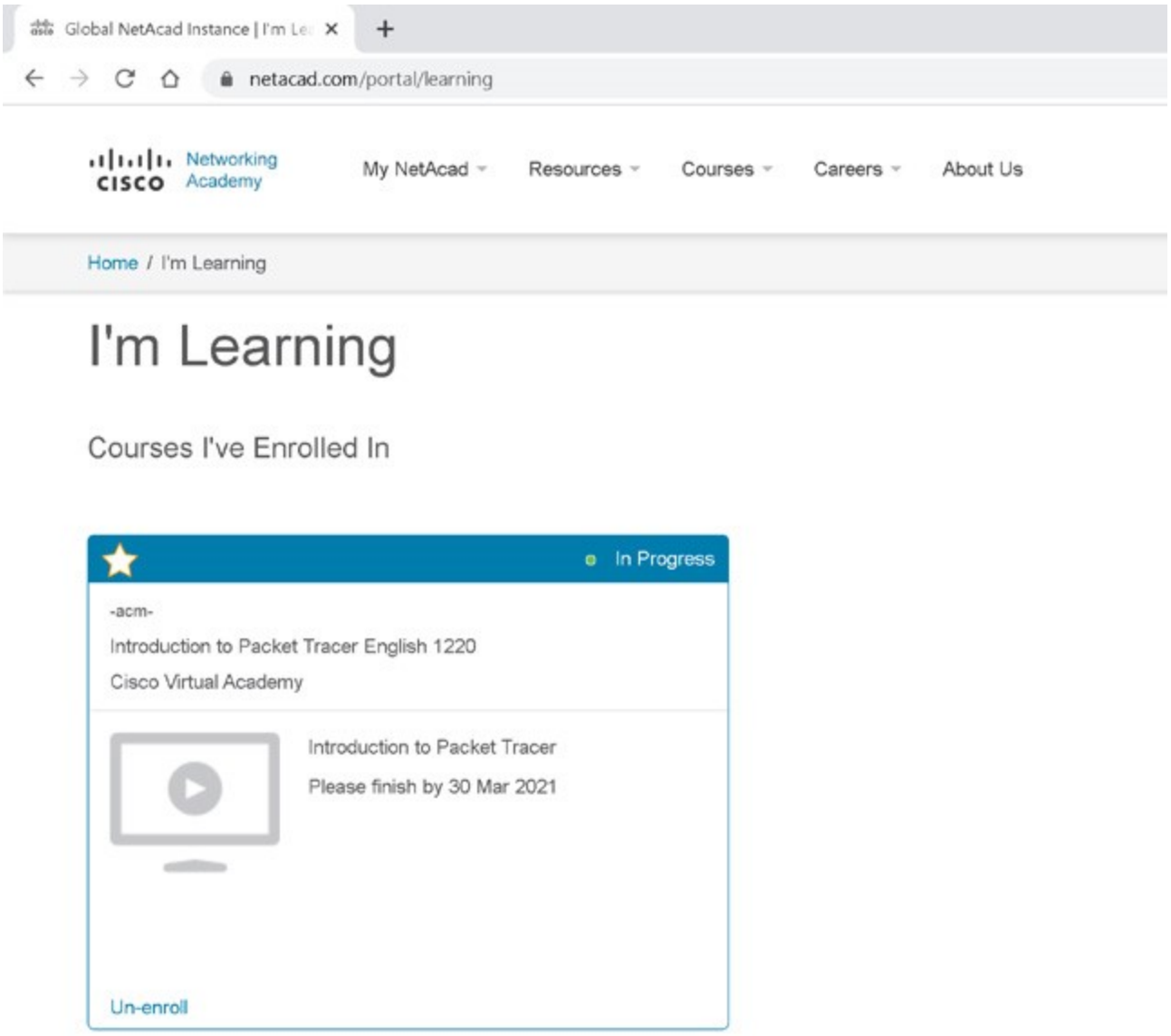


3. Click the **Introduction to Packet Tracer course** hyperlink.

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

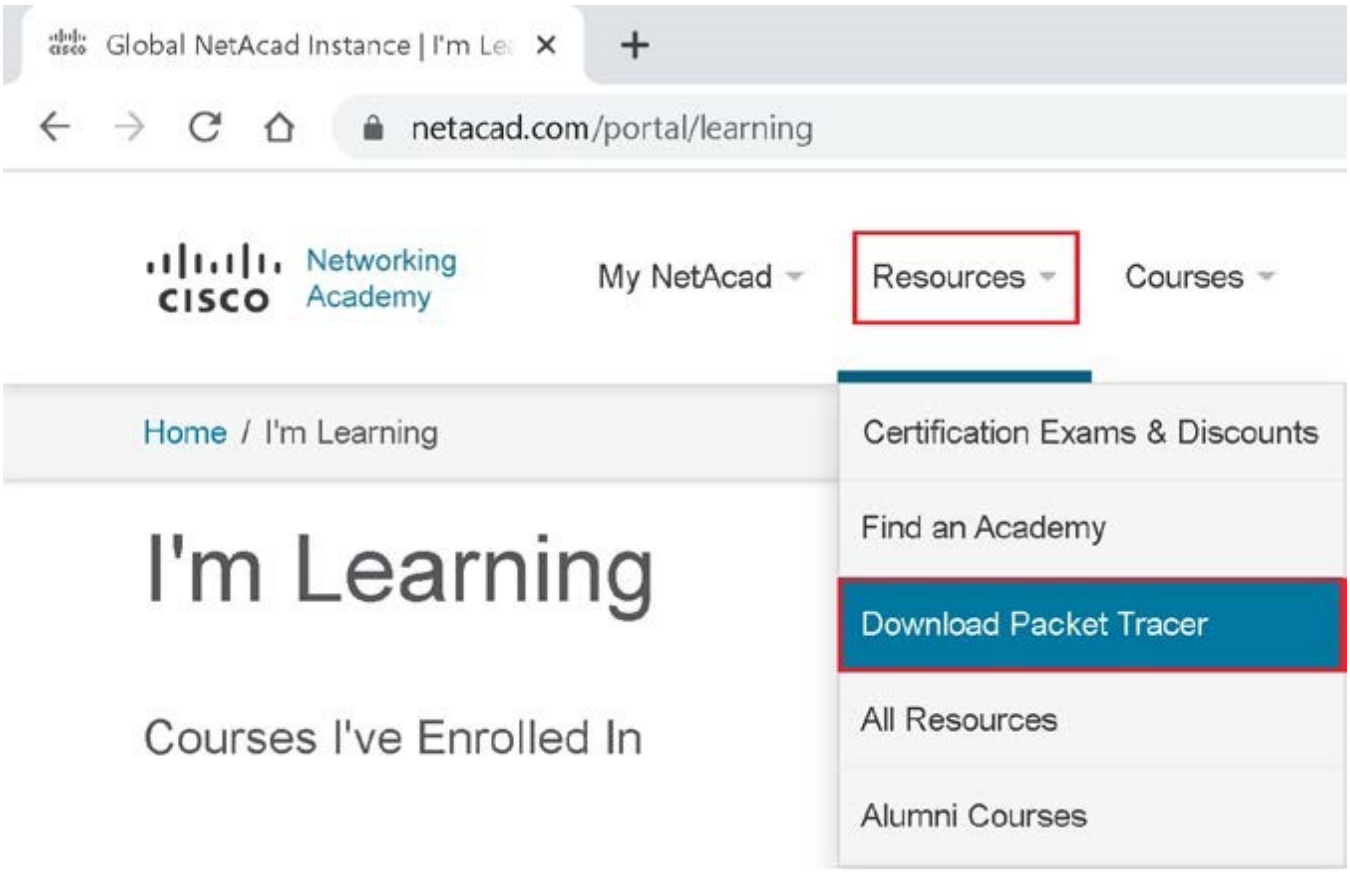


5. Follow the instructions to sign up for the academy.
6. Sign-in to the Academy.

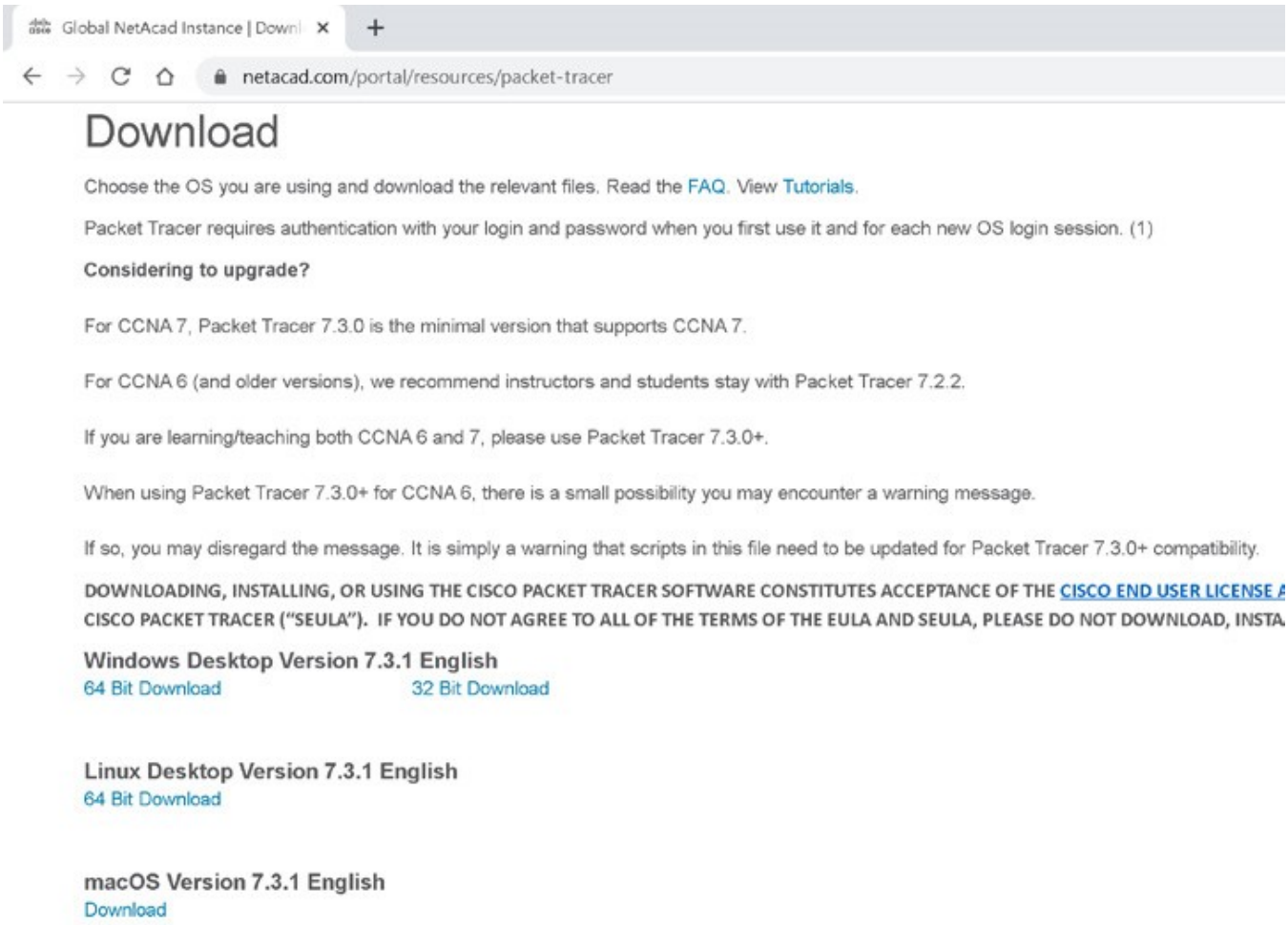


TASK B

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:

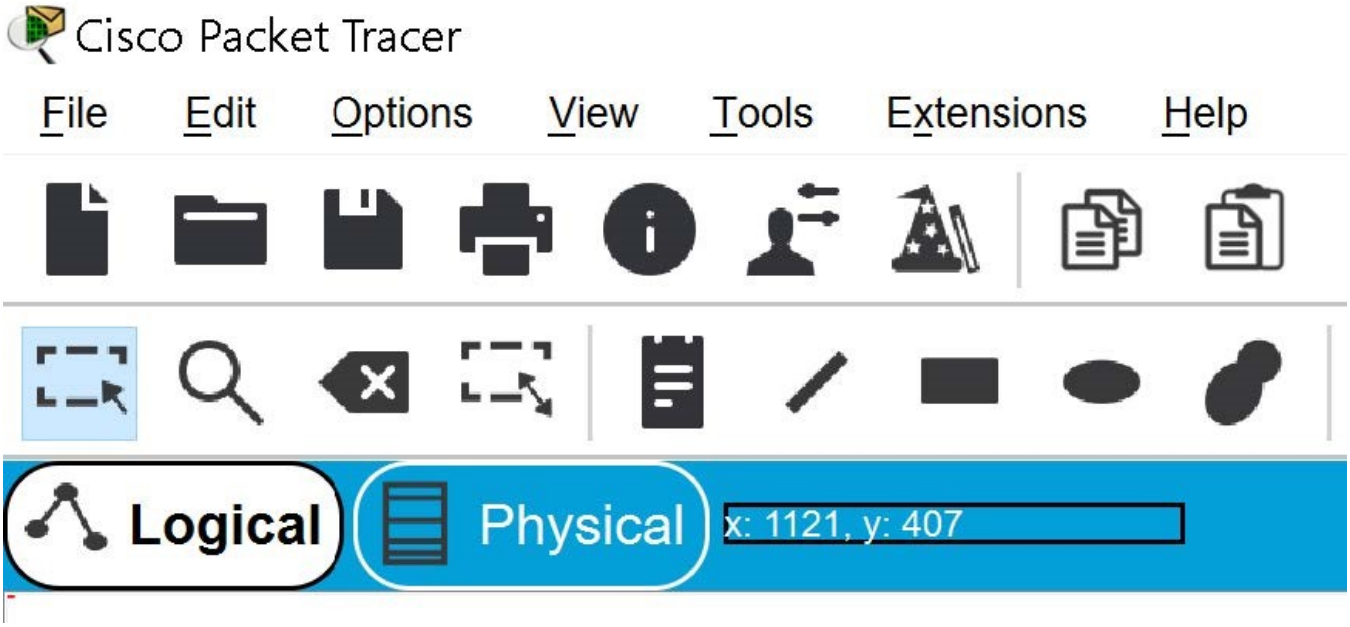
1. Hardware that can put data onto and take data off the medium.
2. A medium to carry data between the devices.
3. Protocols (rules) for how the two devices will communicate.

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

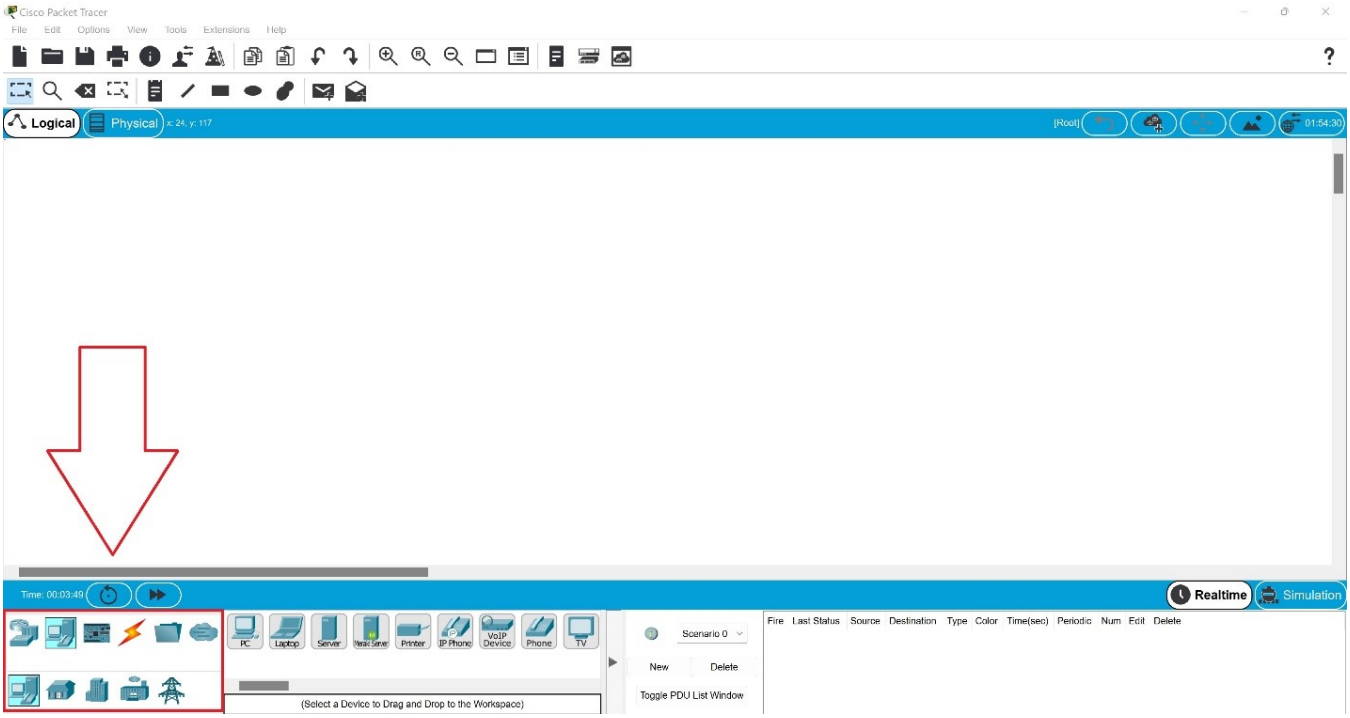
TASK A

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

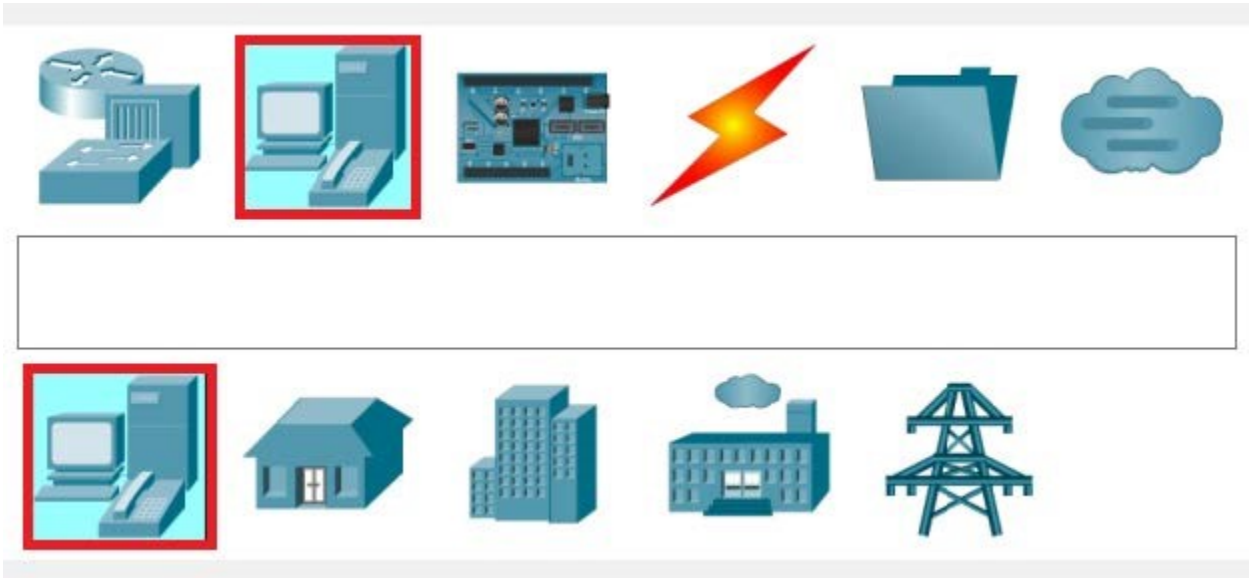
1. If the **Packet Tracer** application is not open, launch **Packet Tracer** and log in if necessary.
2. In the upper left corner, make sure to select the **Logical** button.



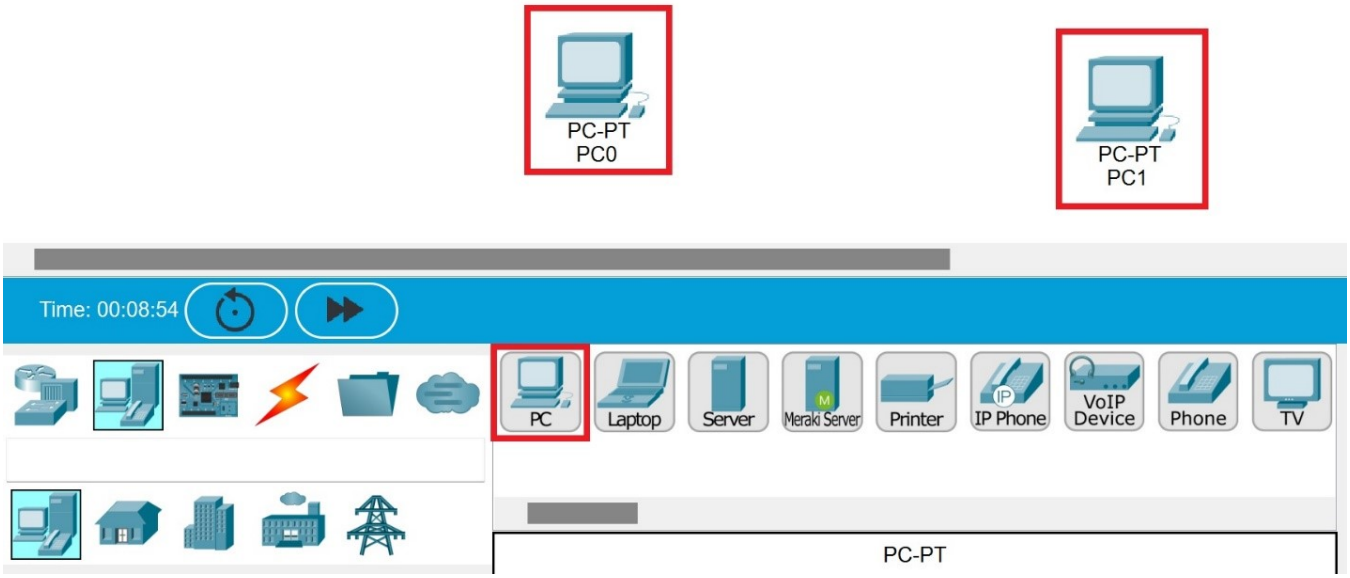
3. In the bottom left corner of the screen, locate the **toolbox area**.



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



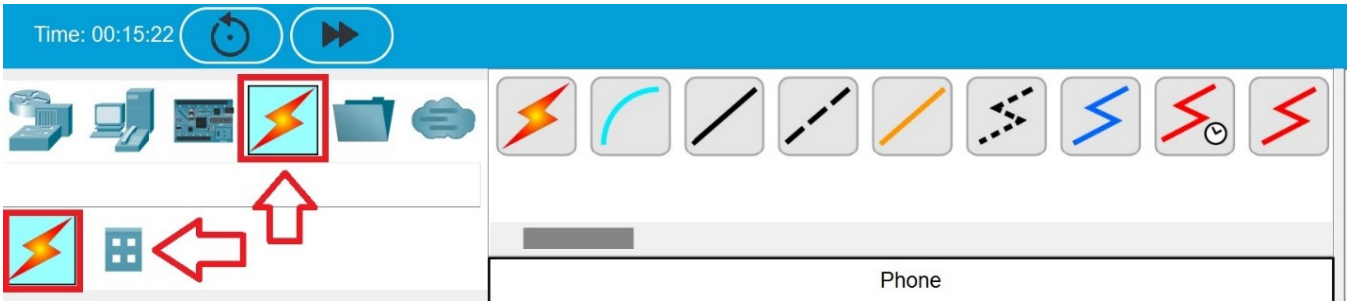
5. Drag two **PC** objects onto the logical diagram.



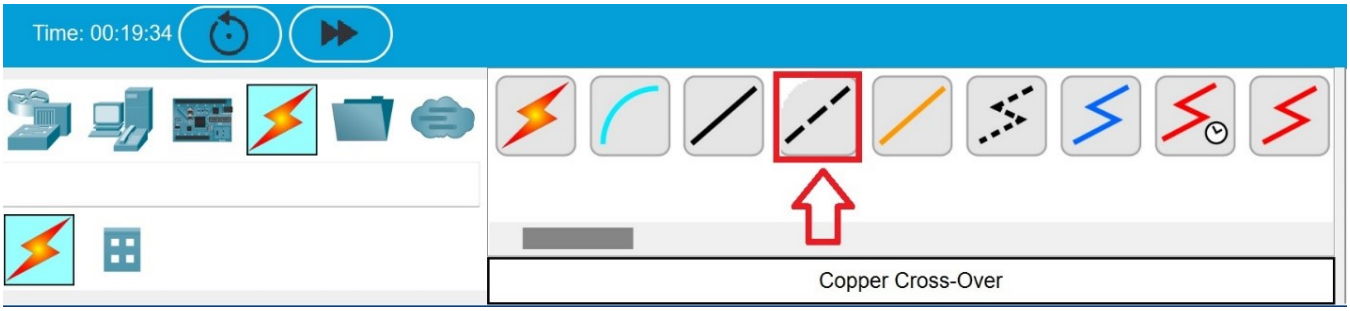
TASK B

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.

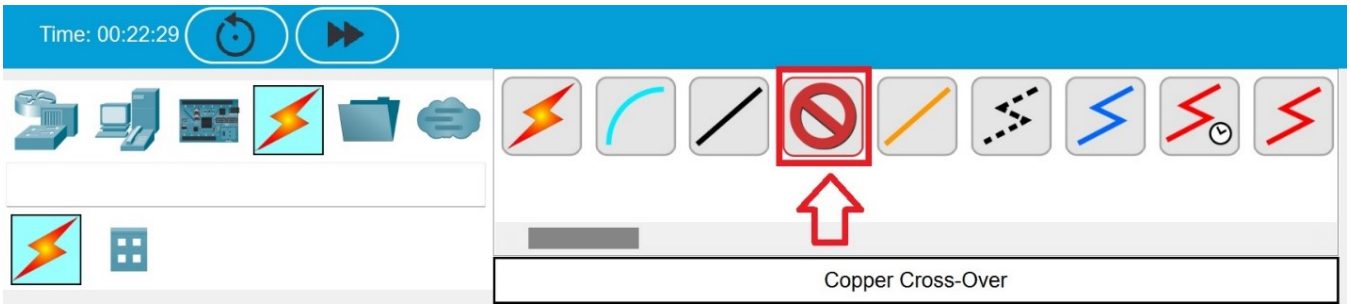
1. In the **toolbox area**, select **Connections** and then select **Connections**.



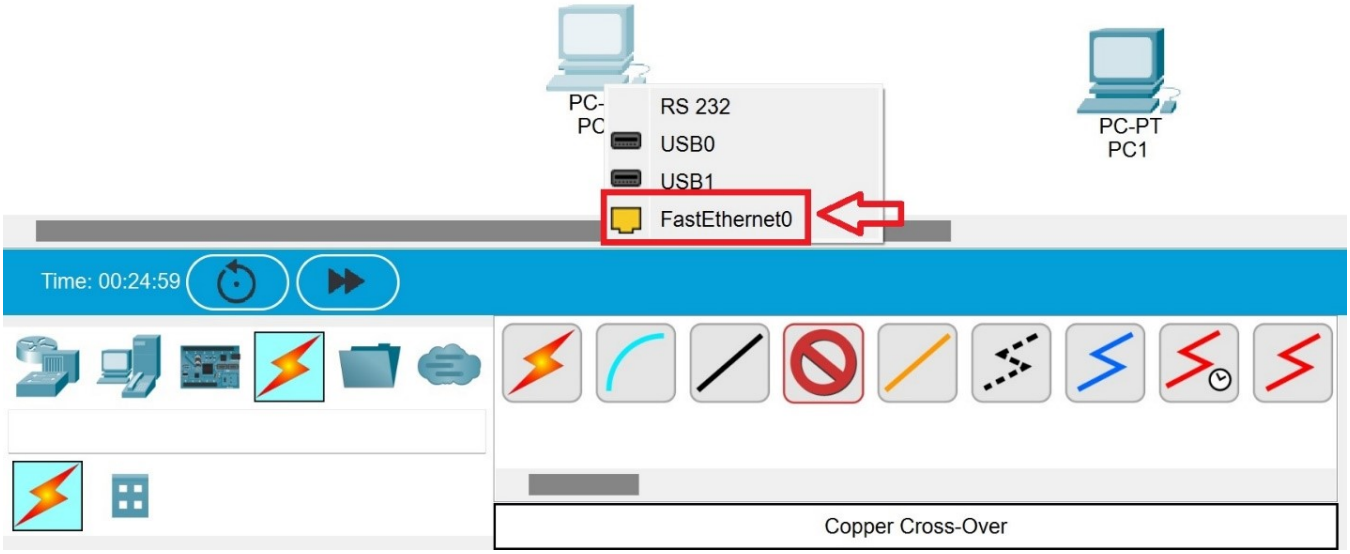
2. Locate the **Copper Cross-Over** cable button.



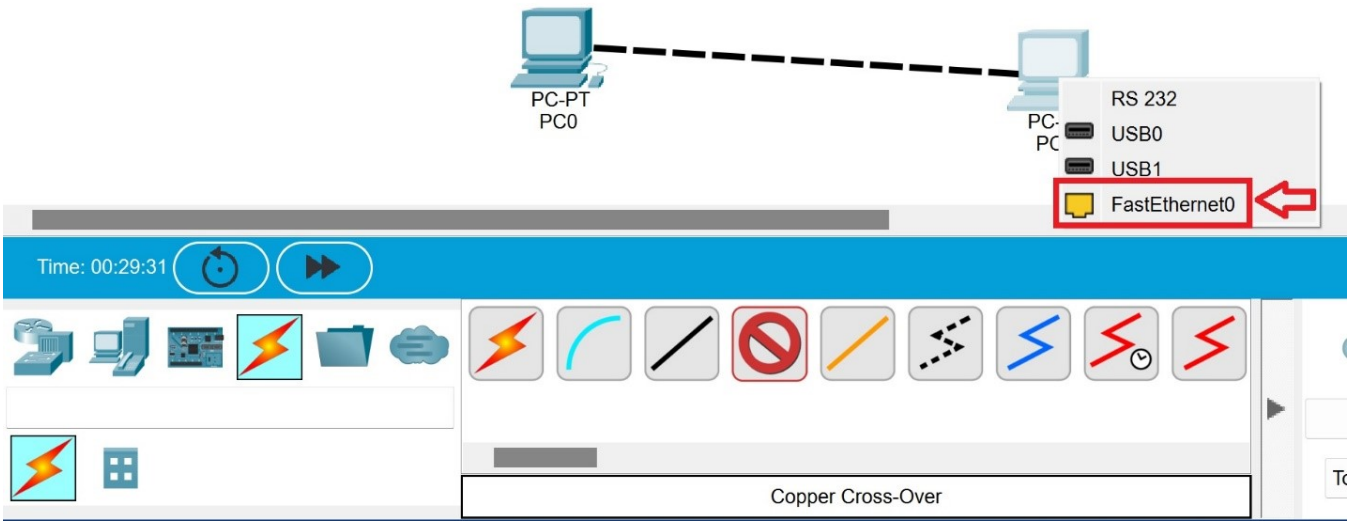
3. Click the **Copper Cross-Over** button (it will change to a red circle with a slash through it.)



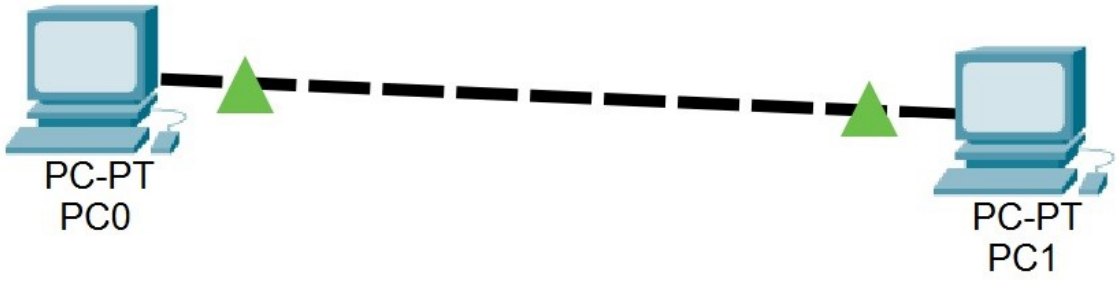
4. Click the **PC-PT PC0** device. A shortcut menu will appear. On the shortcut menu, click **FastEthernet0** to connect one end of the wire to the **PC0** network card.



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

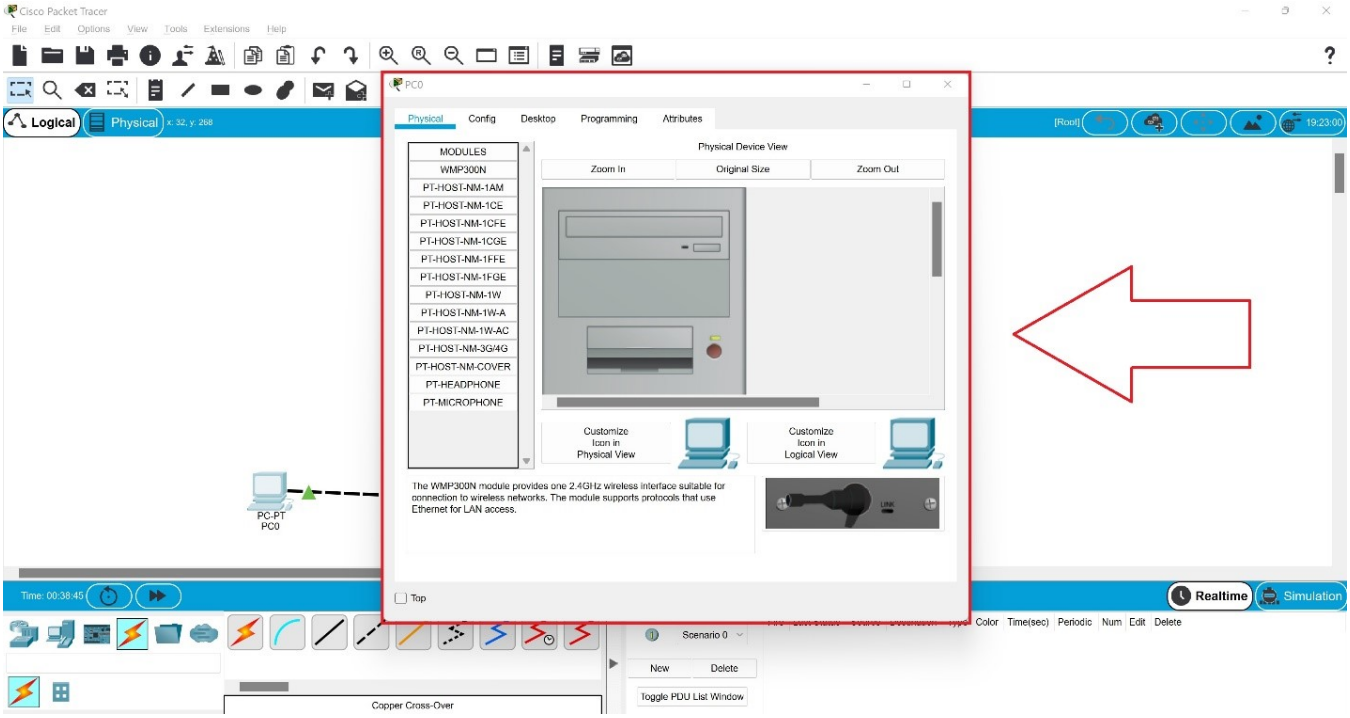
Now that the medium has been connected, we can verify that the computers support the same protocol and then test connectivity between the two nodes.

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.)

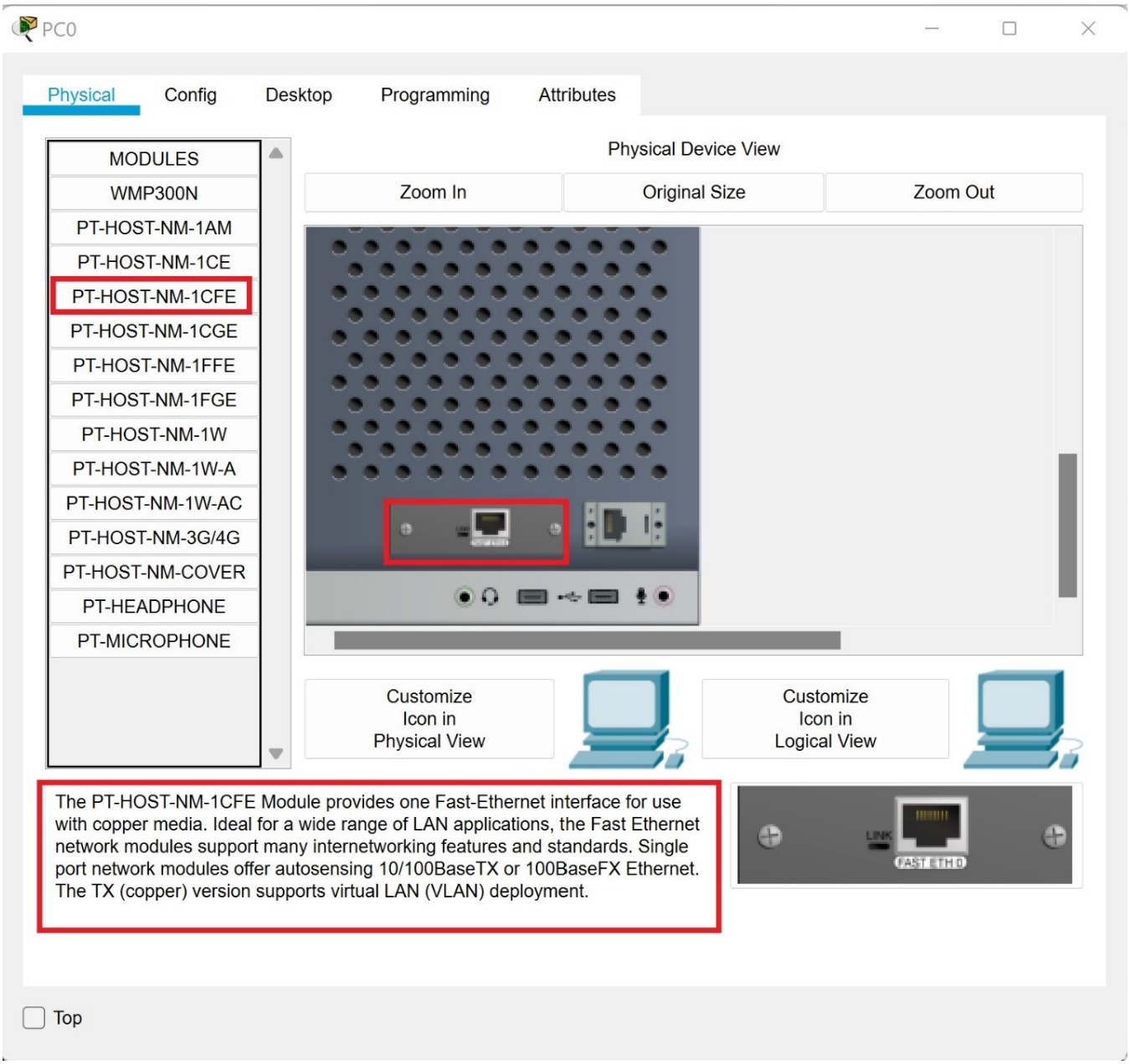
To test connectivity, network administrators commonly use the ping command. This command sends packets to a remote computer which then replies. A successful reply indicates round-trip connectivity.

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

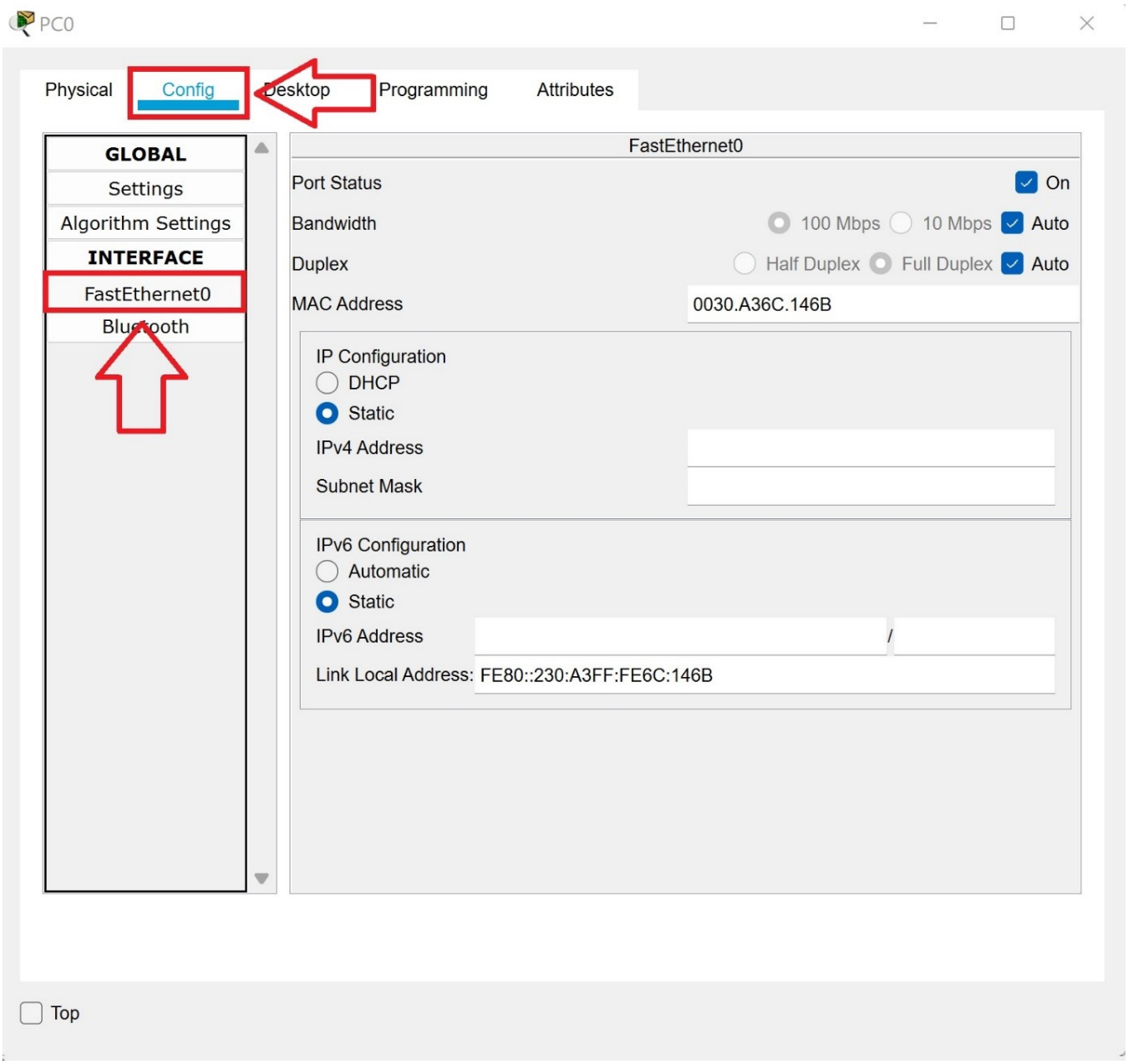
1. Click the **PC-PT PC0** device. The **PC0** dialog box will open.



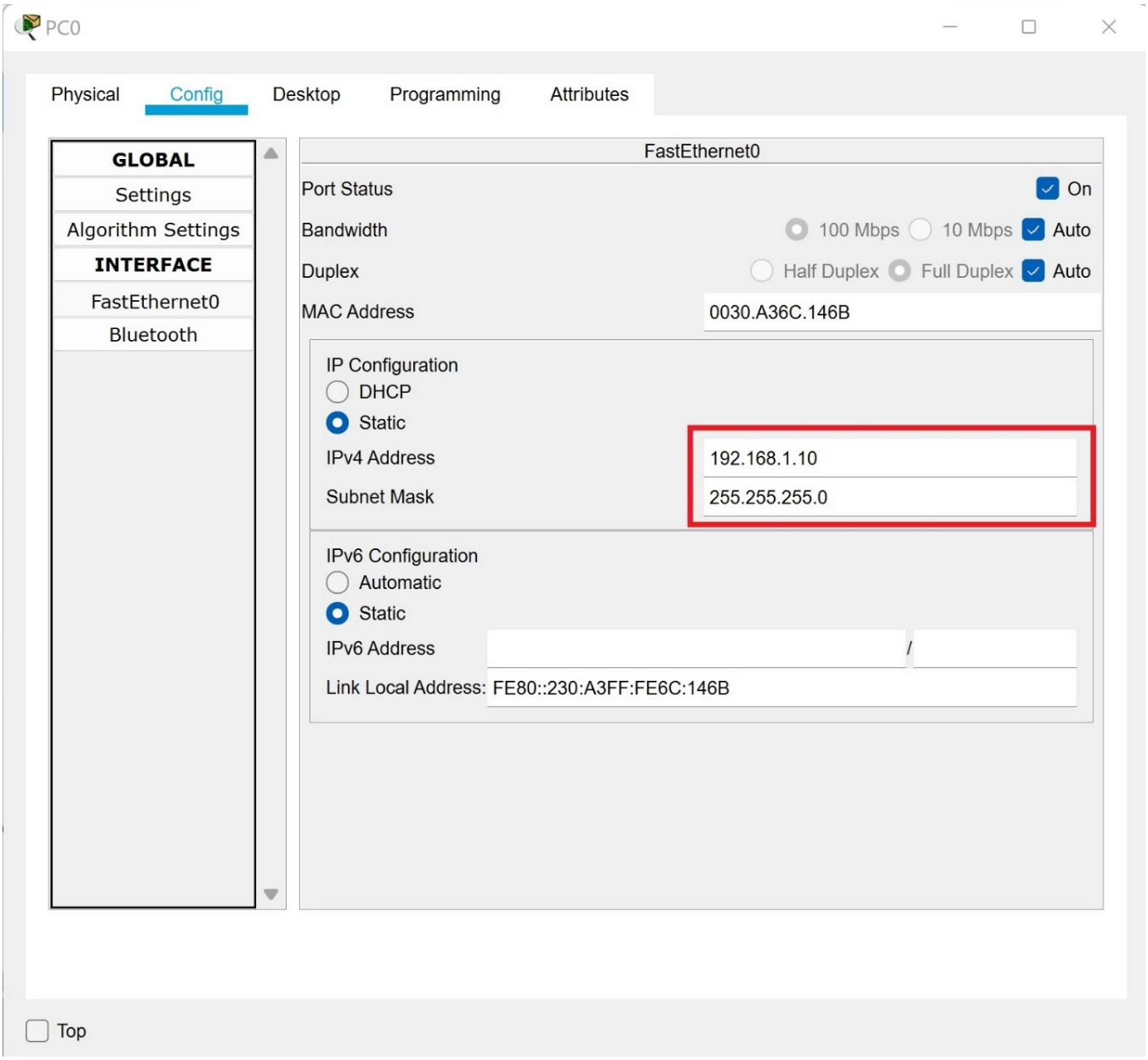
2. In the **Physical Device View** pane, scroll down to view the back of the **PC**. Notice the network card installed in the back of the computer. This is where the copper cross-over cable is connected. The equipment in Packet Tracer work exactly like the equipment would in real life.



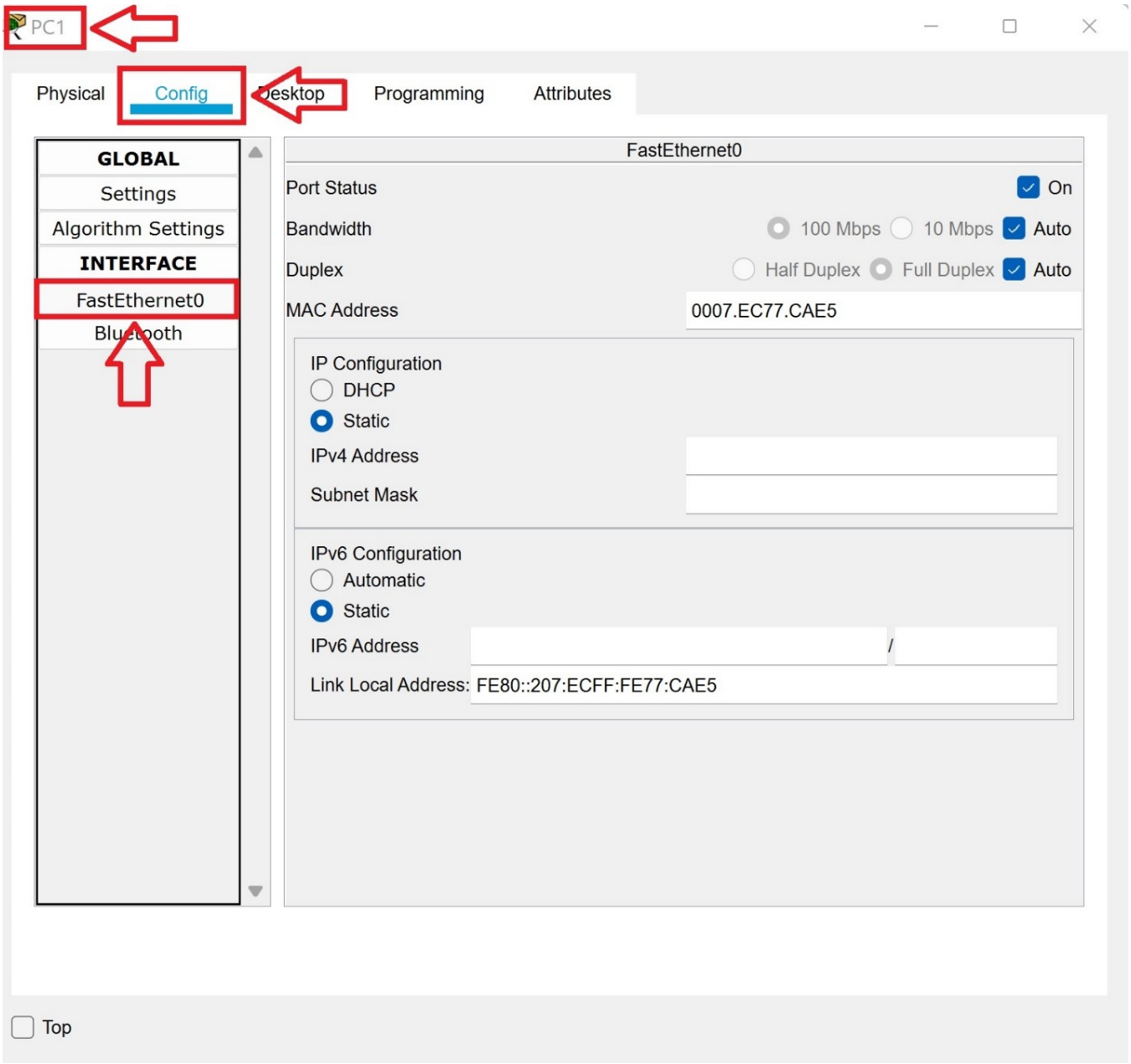
3. In the **PC0** dialog box, click the **Config** tab. Then in the **Interface** menu, click **FastEthernet0** to view the properties of the network interface card (NIC).



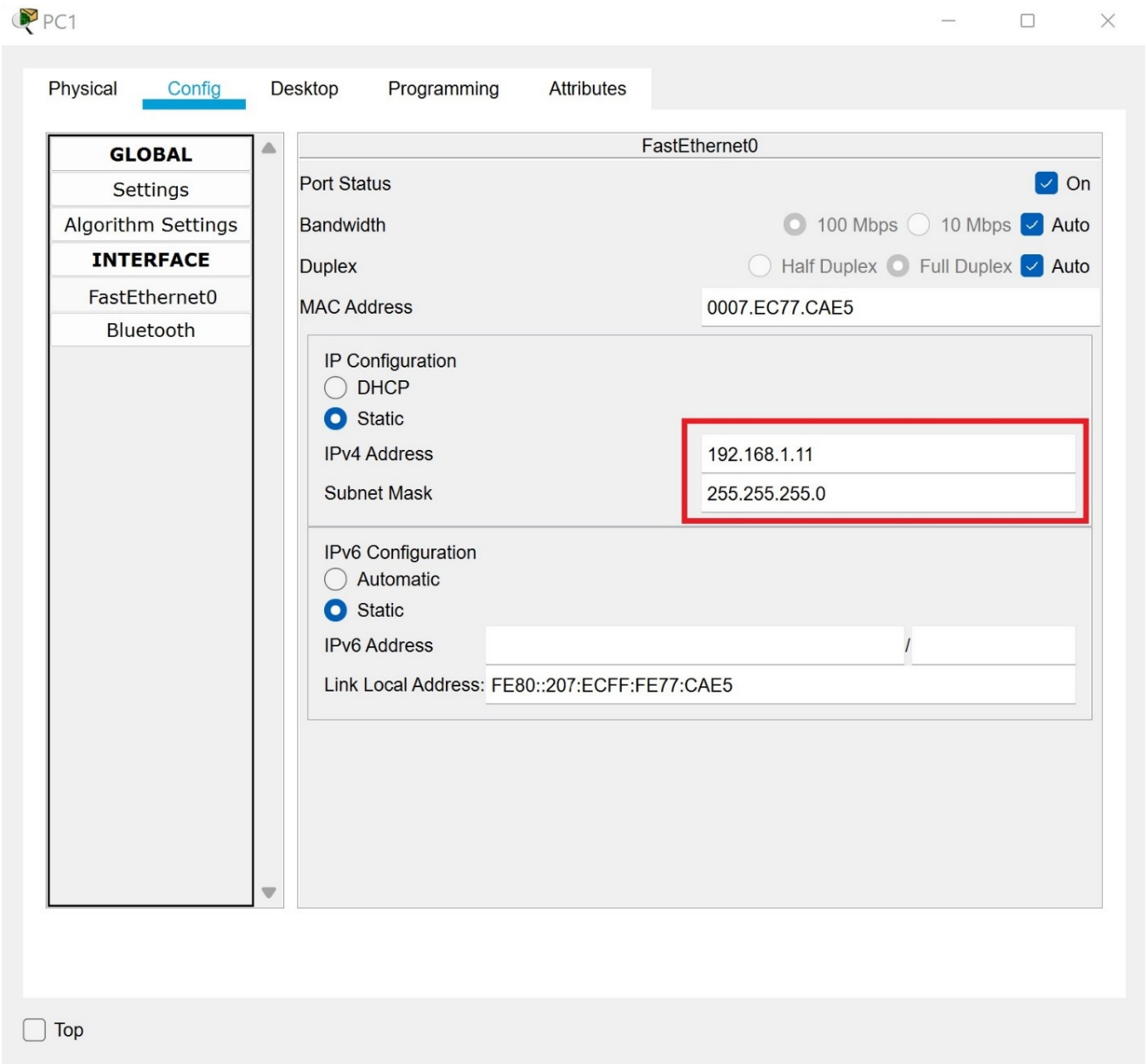
4. In the **IP Configuration** section, click in the **IPv4 Address** text box and enter a static address of **192.168.1.10**. Click in the **Subnet Mask** text box. The software will automatically enter a subnet mask of **255.255.255.0**.



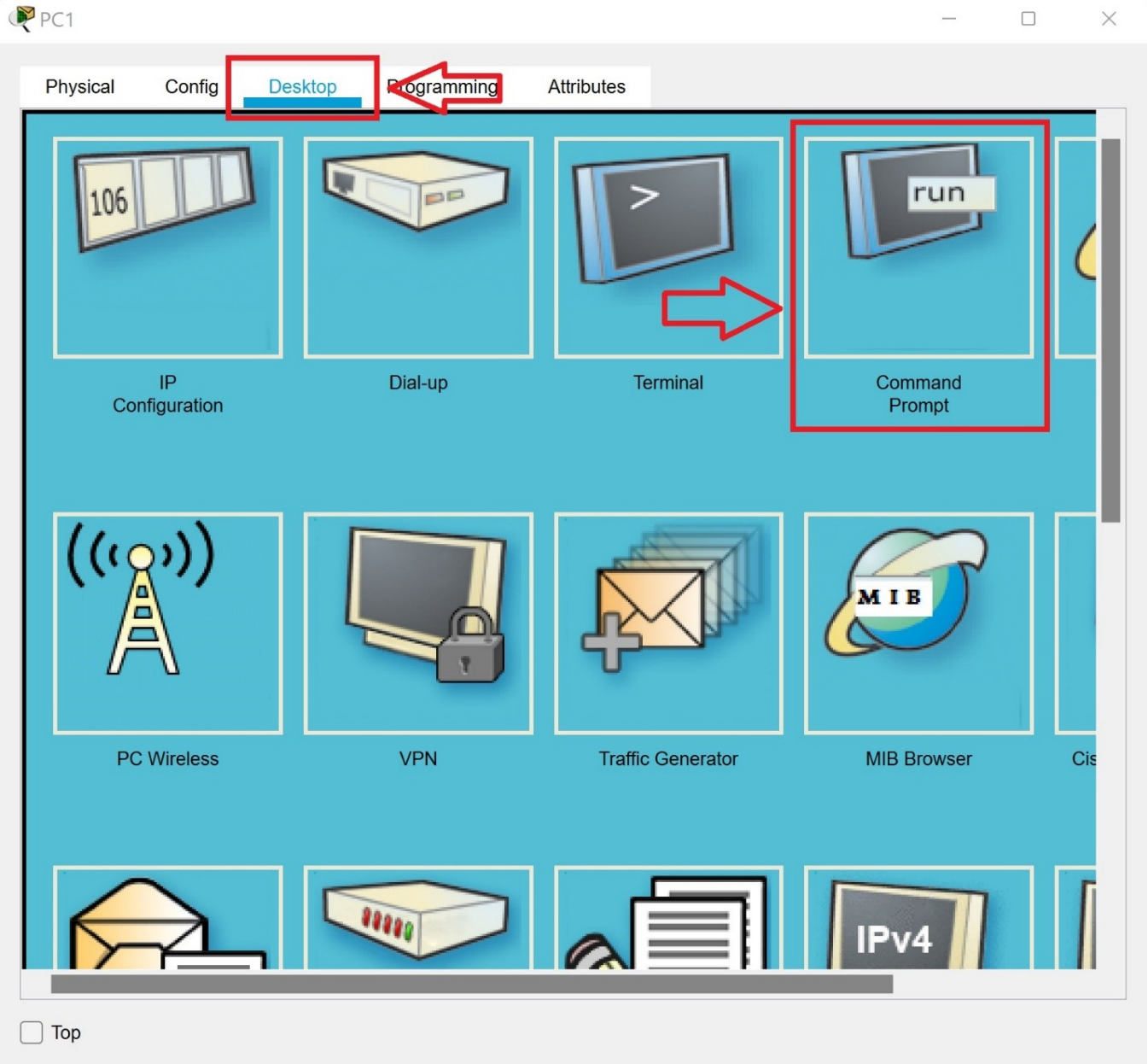
5. Close the **PC0** dialog box.
6. Click the **PC-PT PC1** device. The **PC1** dialog box will open. Click the **Config** tab. Then in the **Interface** menu, click **FastEthernet0** to view the properties of the network interface card (NIC).



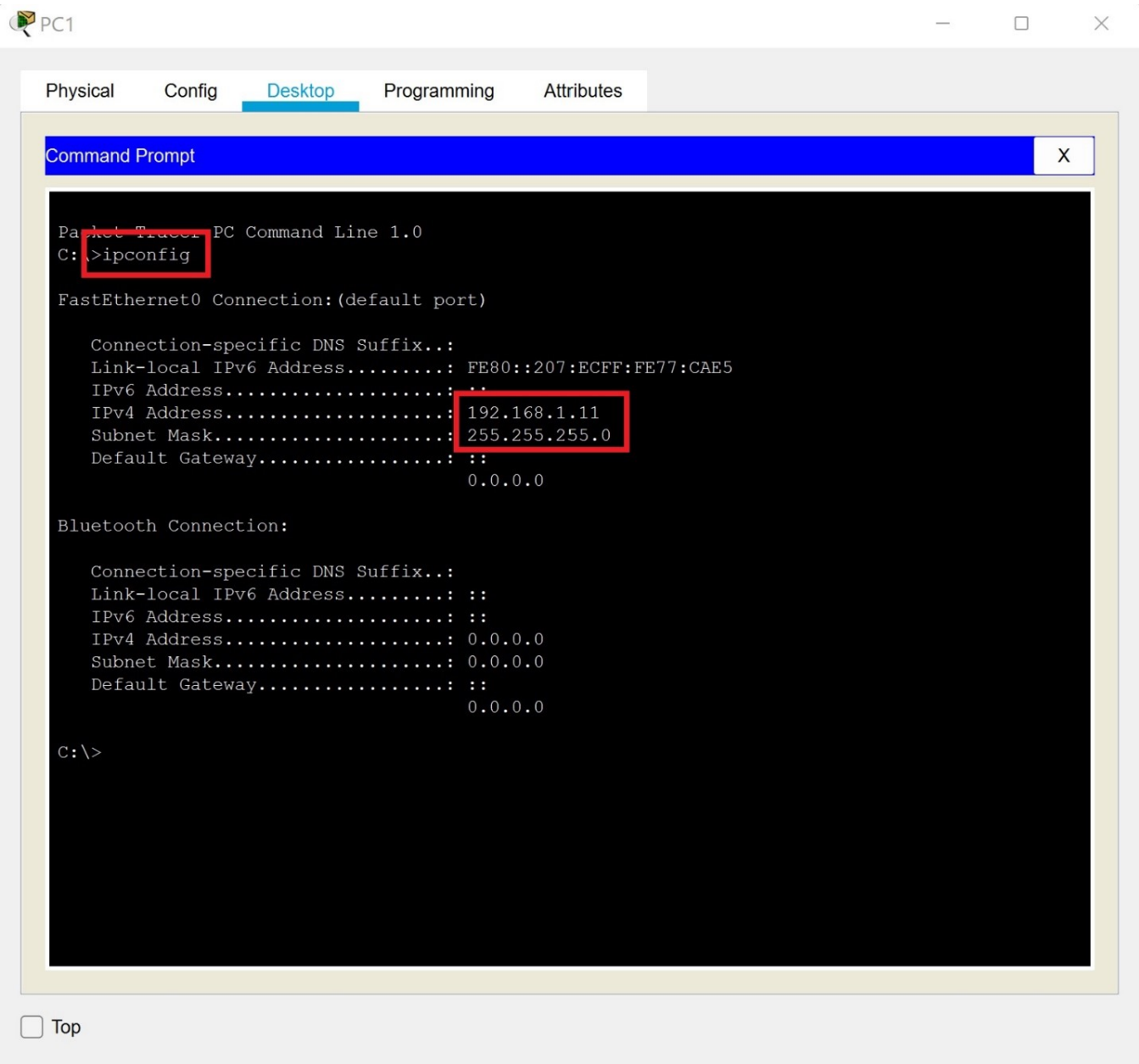
7. In the **IP Configuration** section, click in the **IPv4 Address** text box and enter a static address of **192.168.1.11**. Click in the **Subnet Mask** text box. The software will automatically enter a subnet mask of **255.255.255.0**.



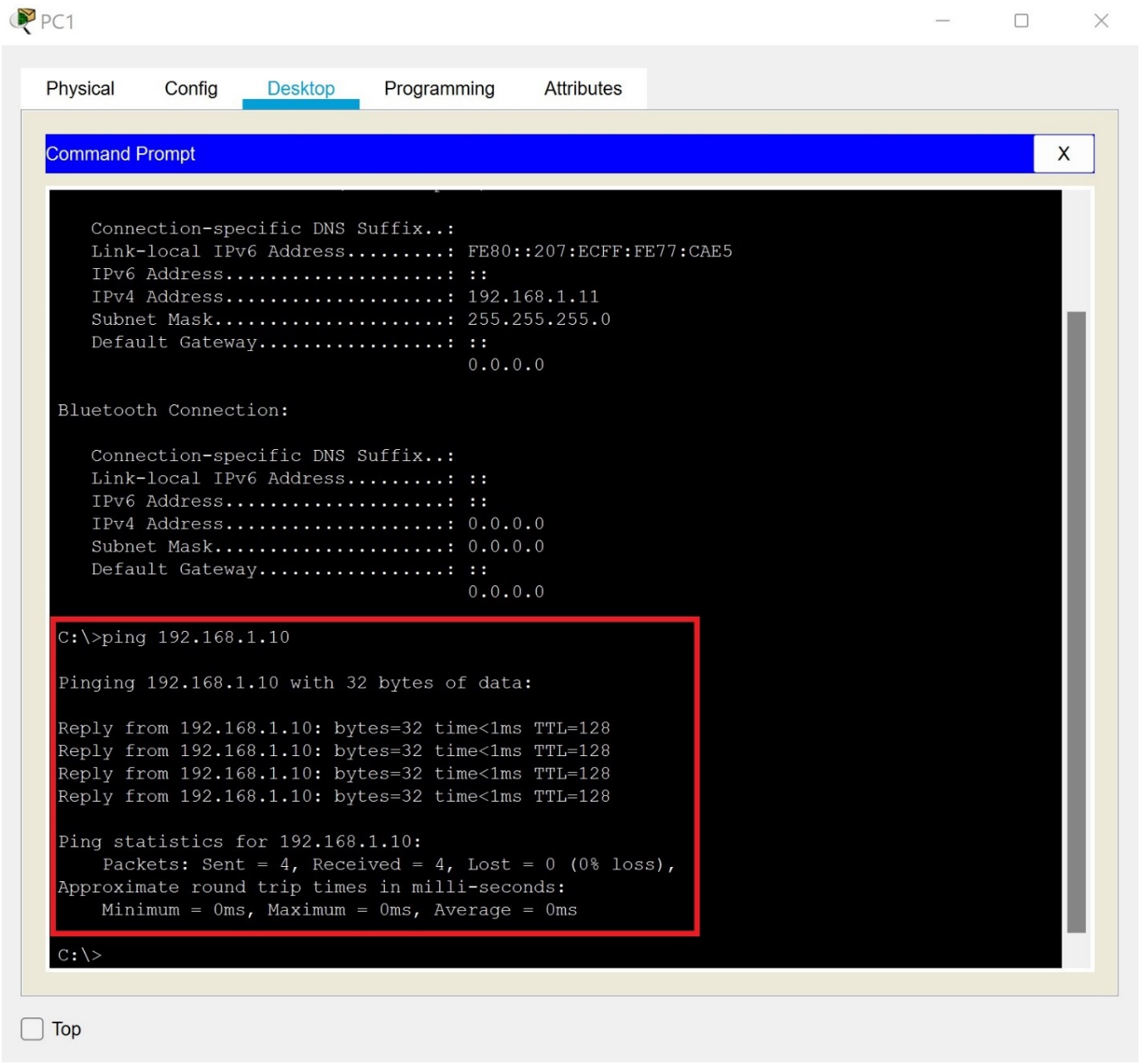
8. In the **PC1** dialog box, click the **Desktop** tab. Then, click the **Command Prompt** option.



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.

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

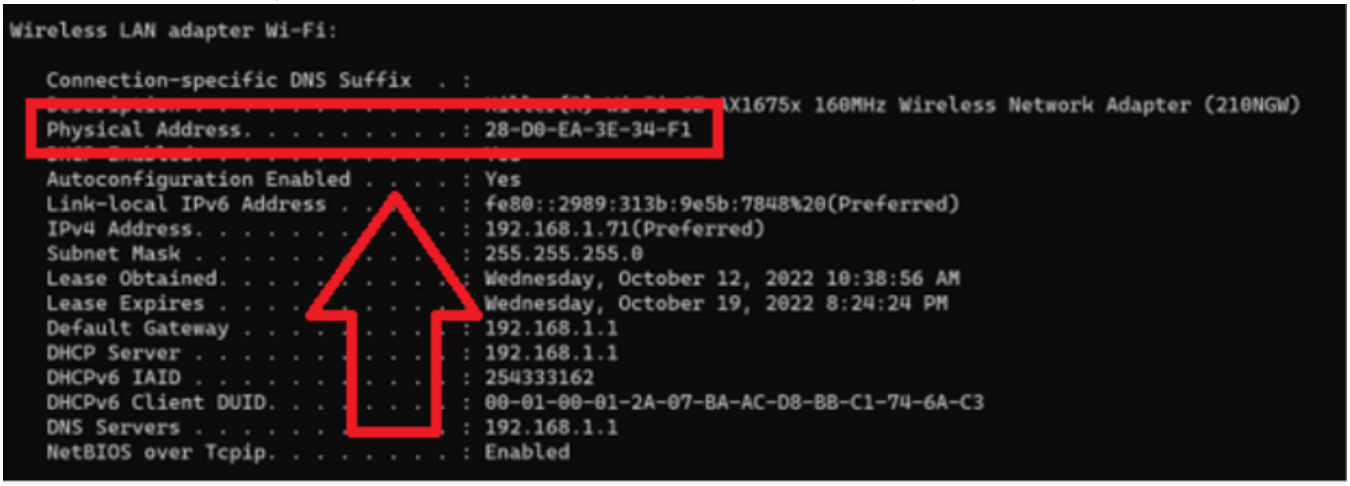
Network Interface Cards (NICs) connect devices to the network. Network adapter or network card are both alternate names for NICs. The NIC serves as an interface between a computer and the network. To connect to a network, a computer must have a NIC installed.
NICs can be built into the motherboard of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

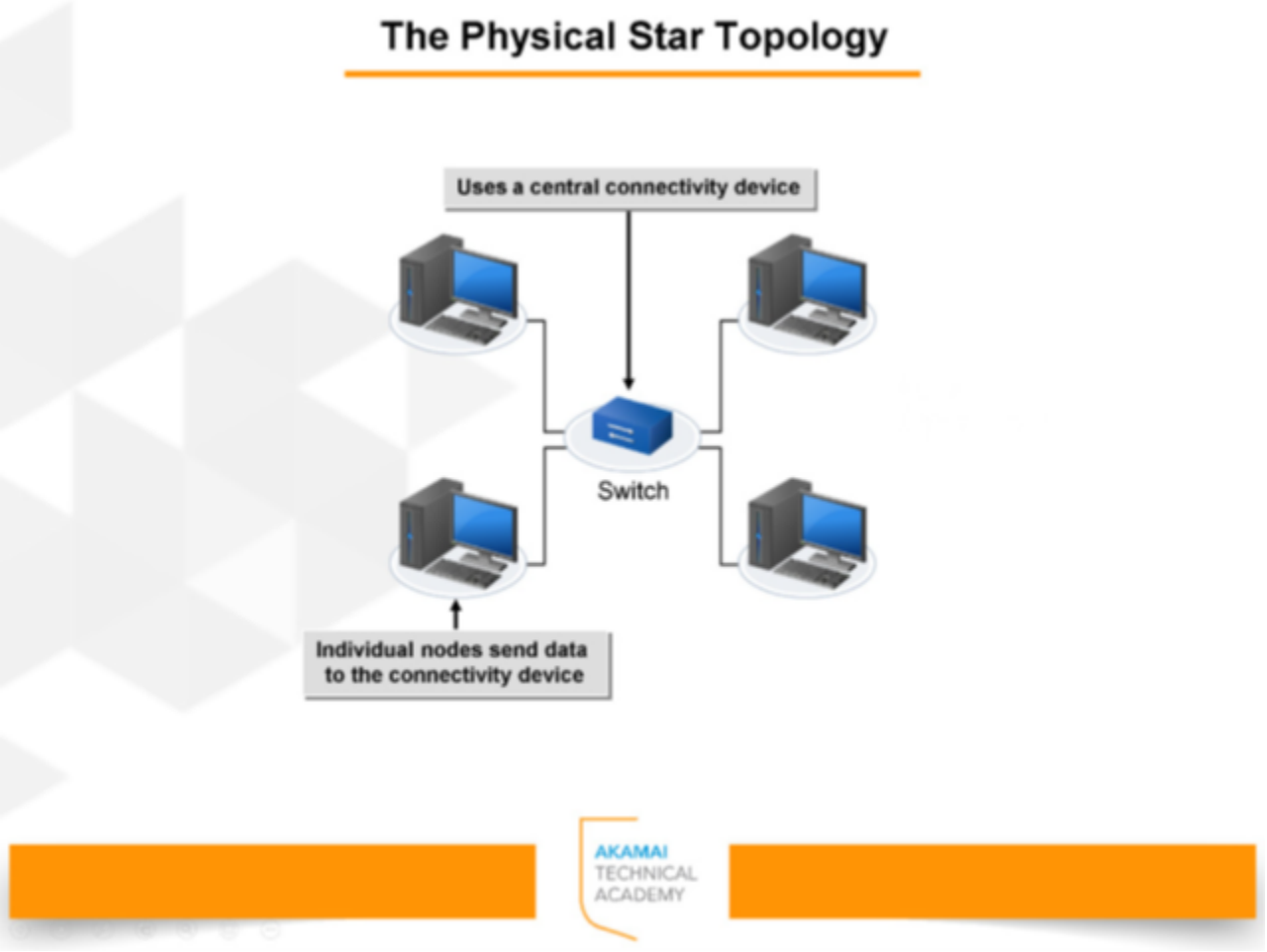
To deliver something like mail or data, the recipient must have a unique address. Imagine if there were two houses that had the same address. How would the mail system know where to deliver each letter or package?
The same is true for NICs. Each NIC must have a unique address. That address is called 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.

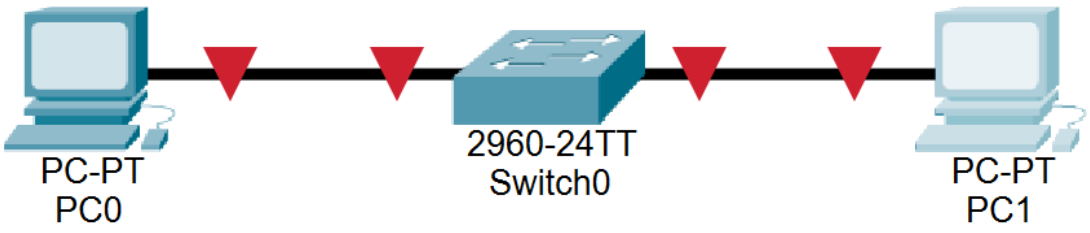
Network Transmissions and hardware Lab

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.

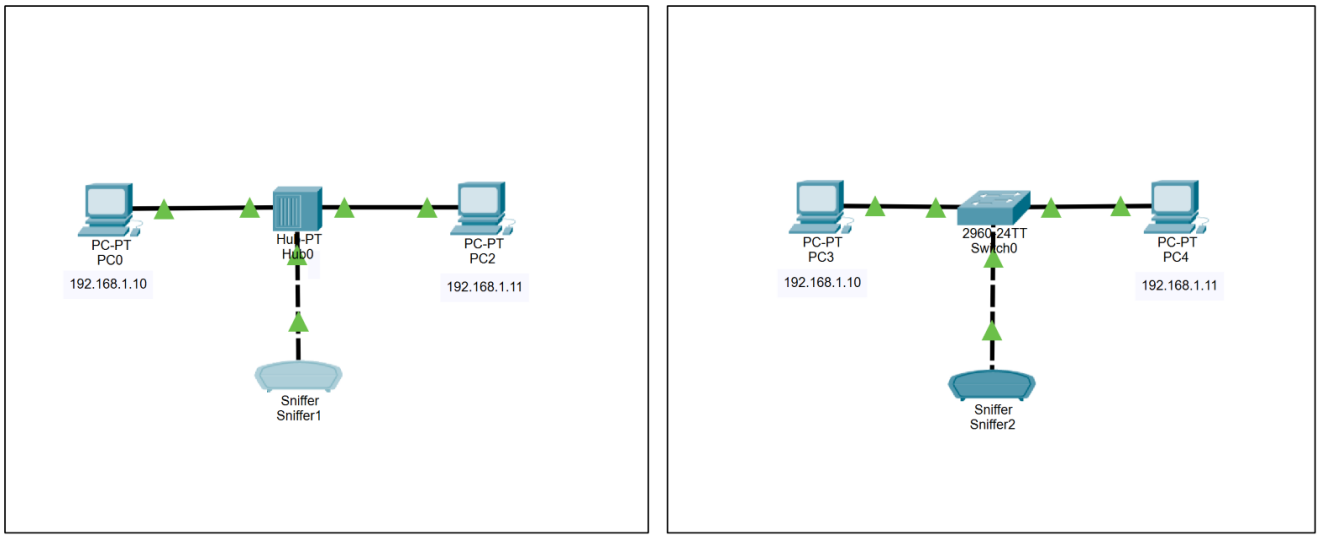
TASK A

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

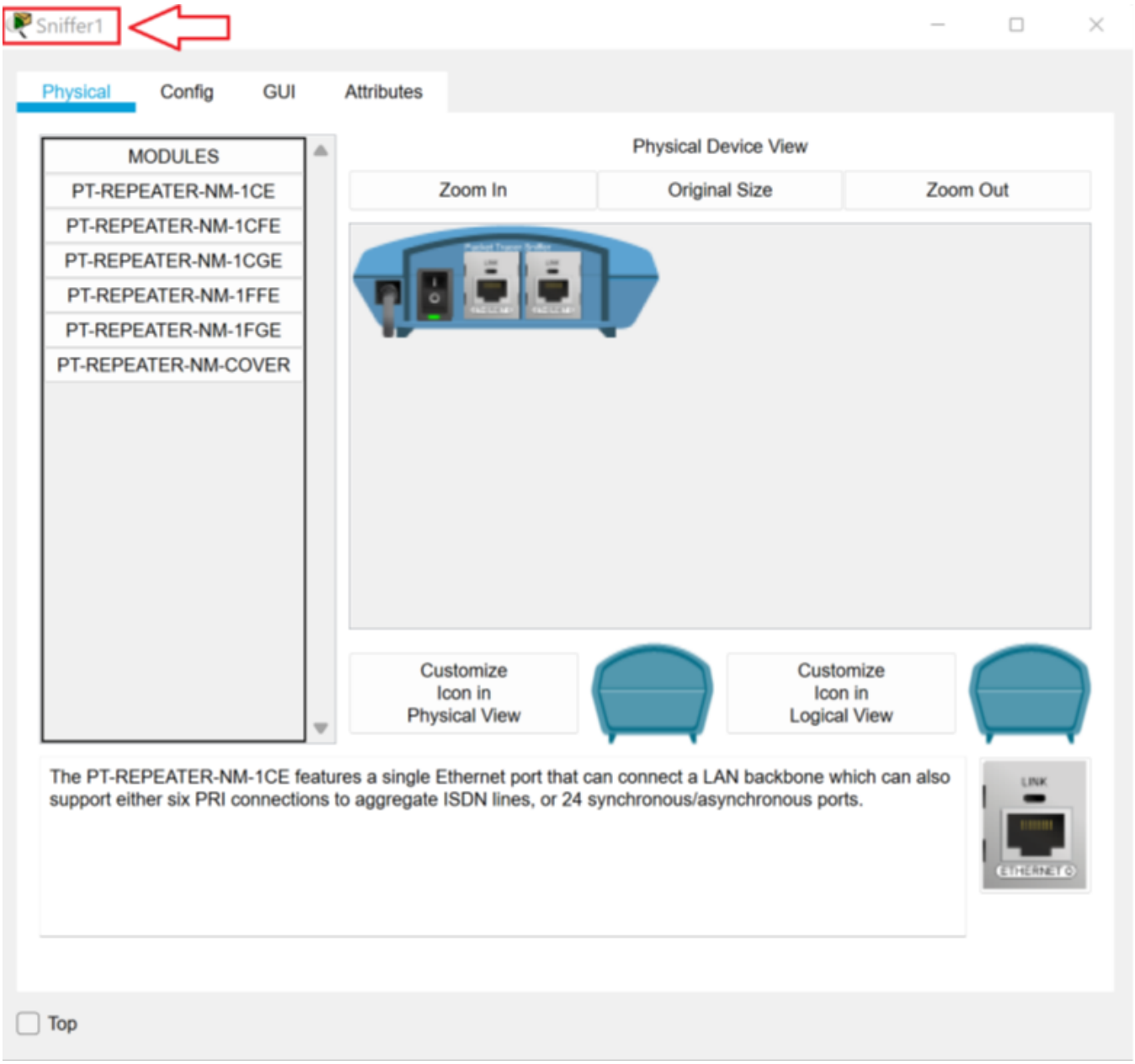
1. Download the **3.4.2 Lab File** and open it in **Packet Tracer**.

3.4.2 Lab File

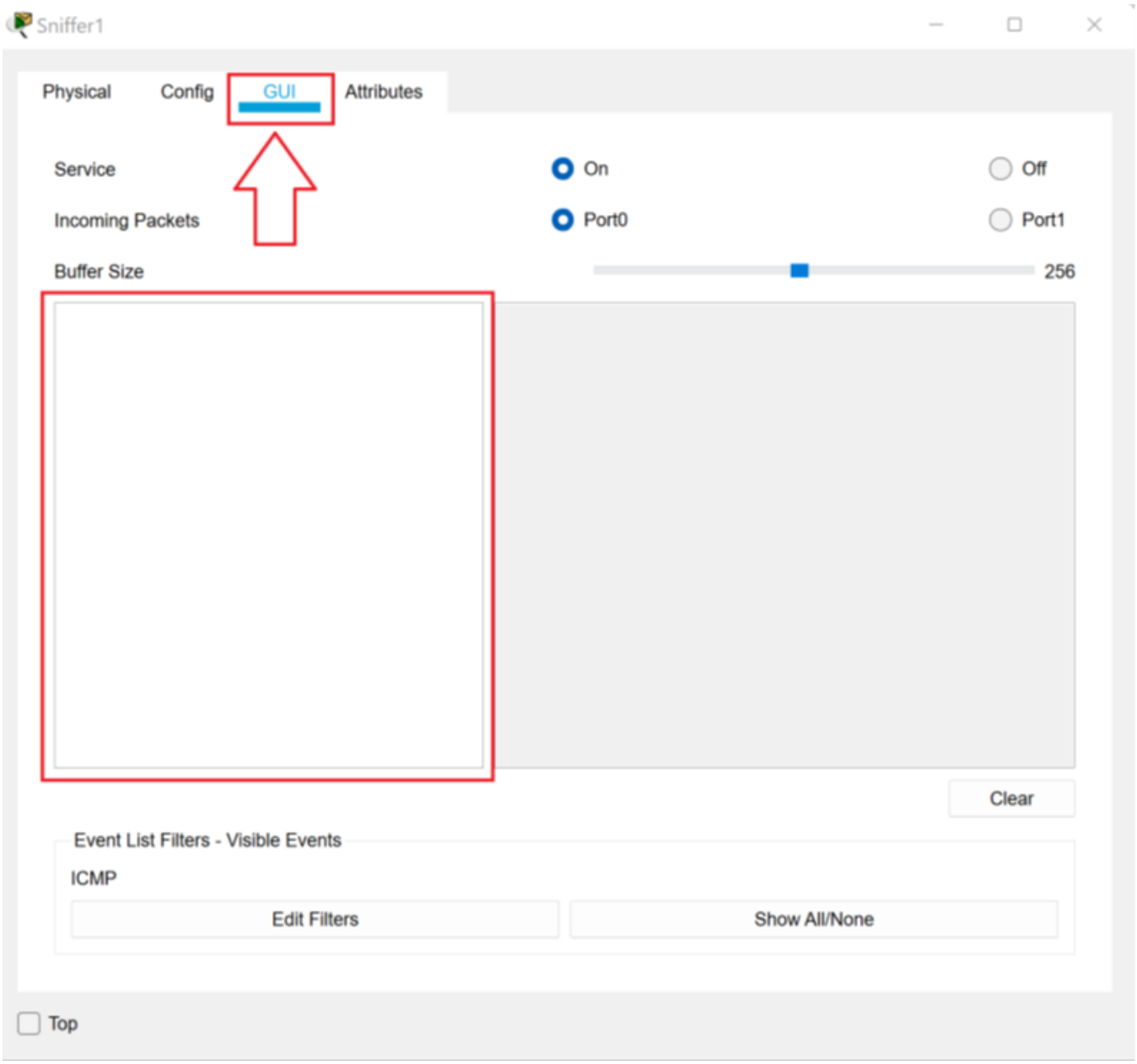
PKT File



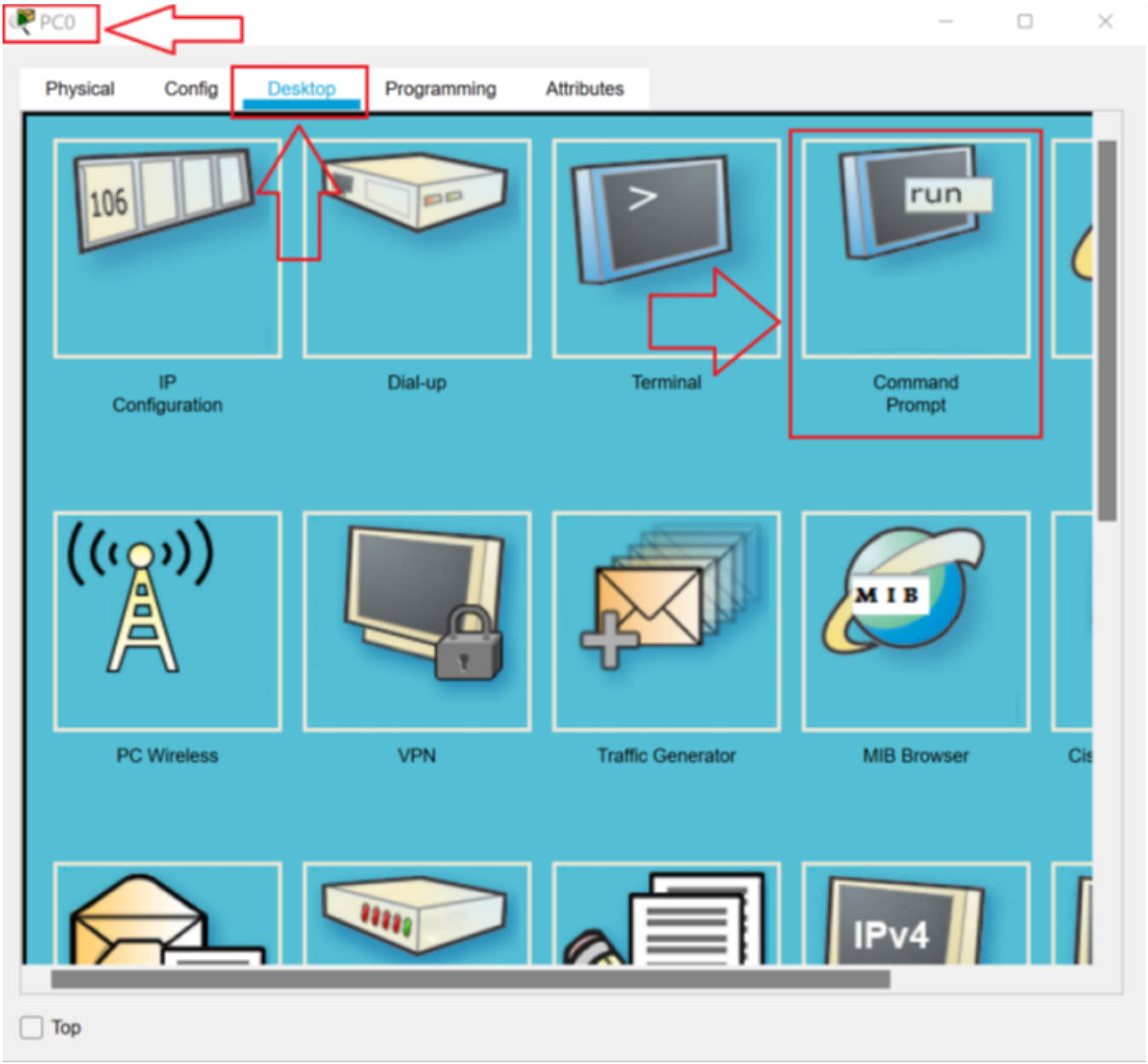
2. Click on **Sniffer1** to open the **Sniffer1 Properties** dialog box.



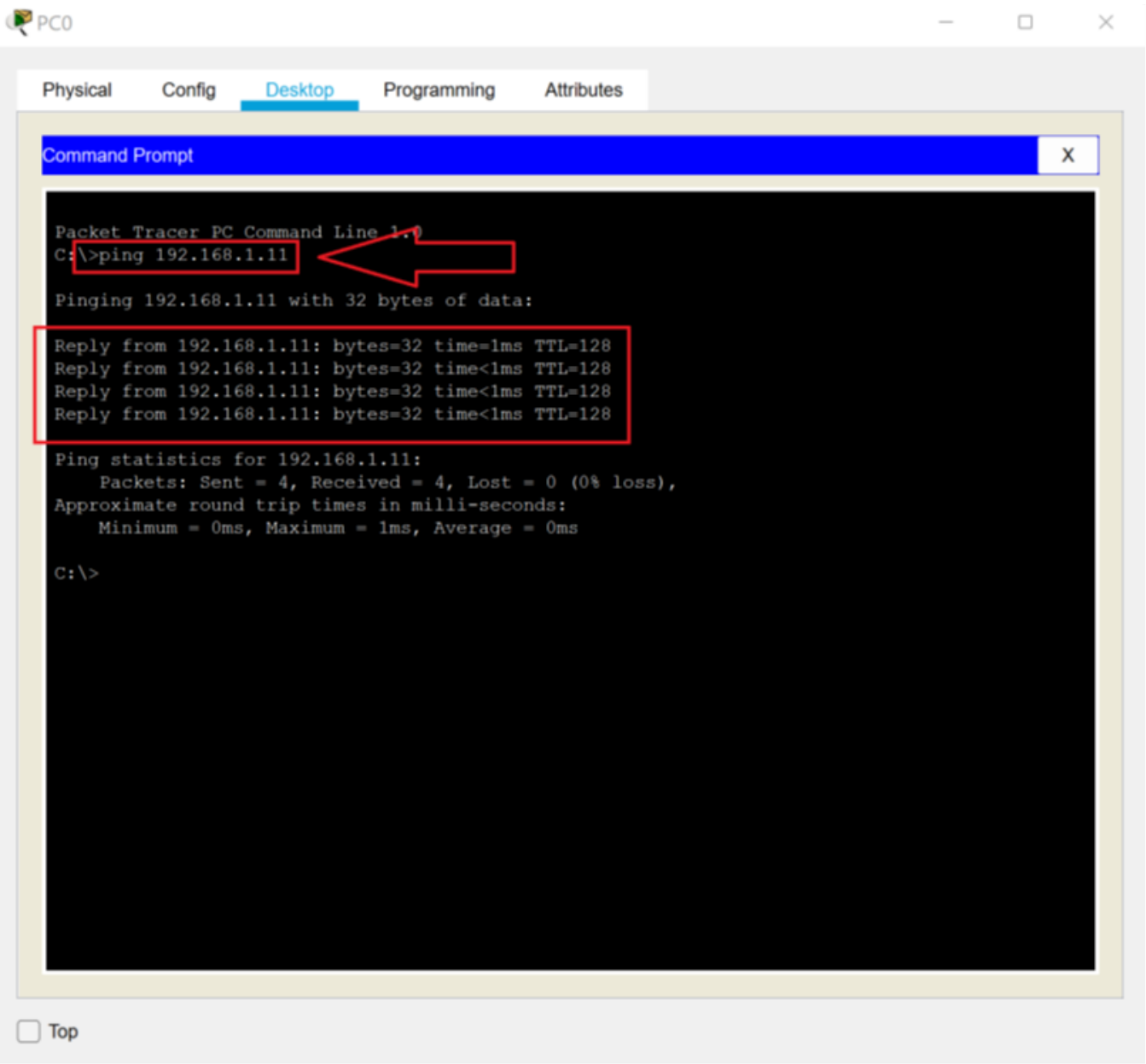
3. Click on the **GUI** tab. (NOTE: If there are any events in the buffer, click the **Clear** button to clear them.)



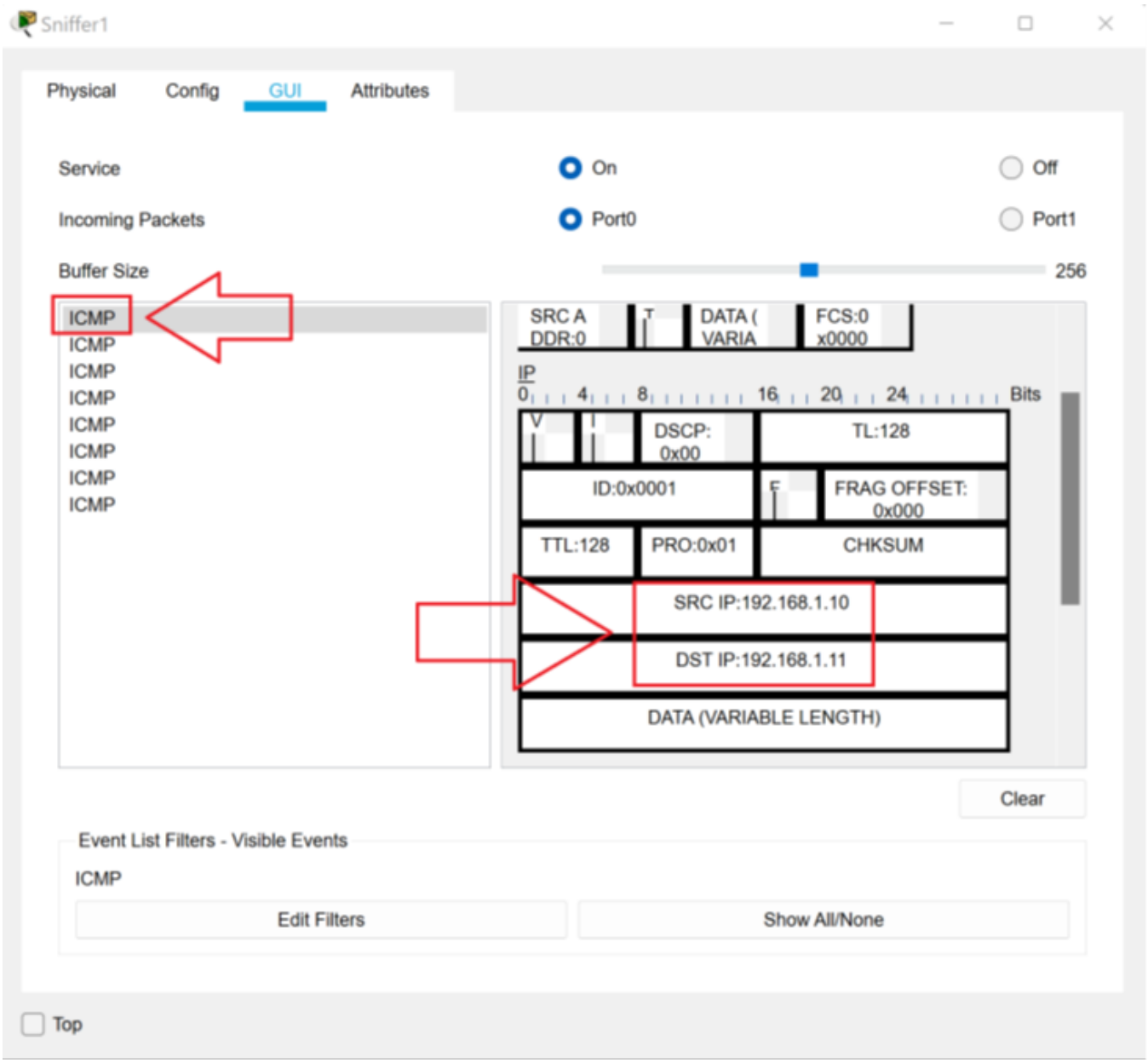
4. Close the **Sniffer1 Properties** dialog box.
5. Click **PC0** to open the **PC0 Properties** dialog box. Then click the **Desktop** tab.
6. Click the **Command Prompt** icon.



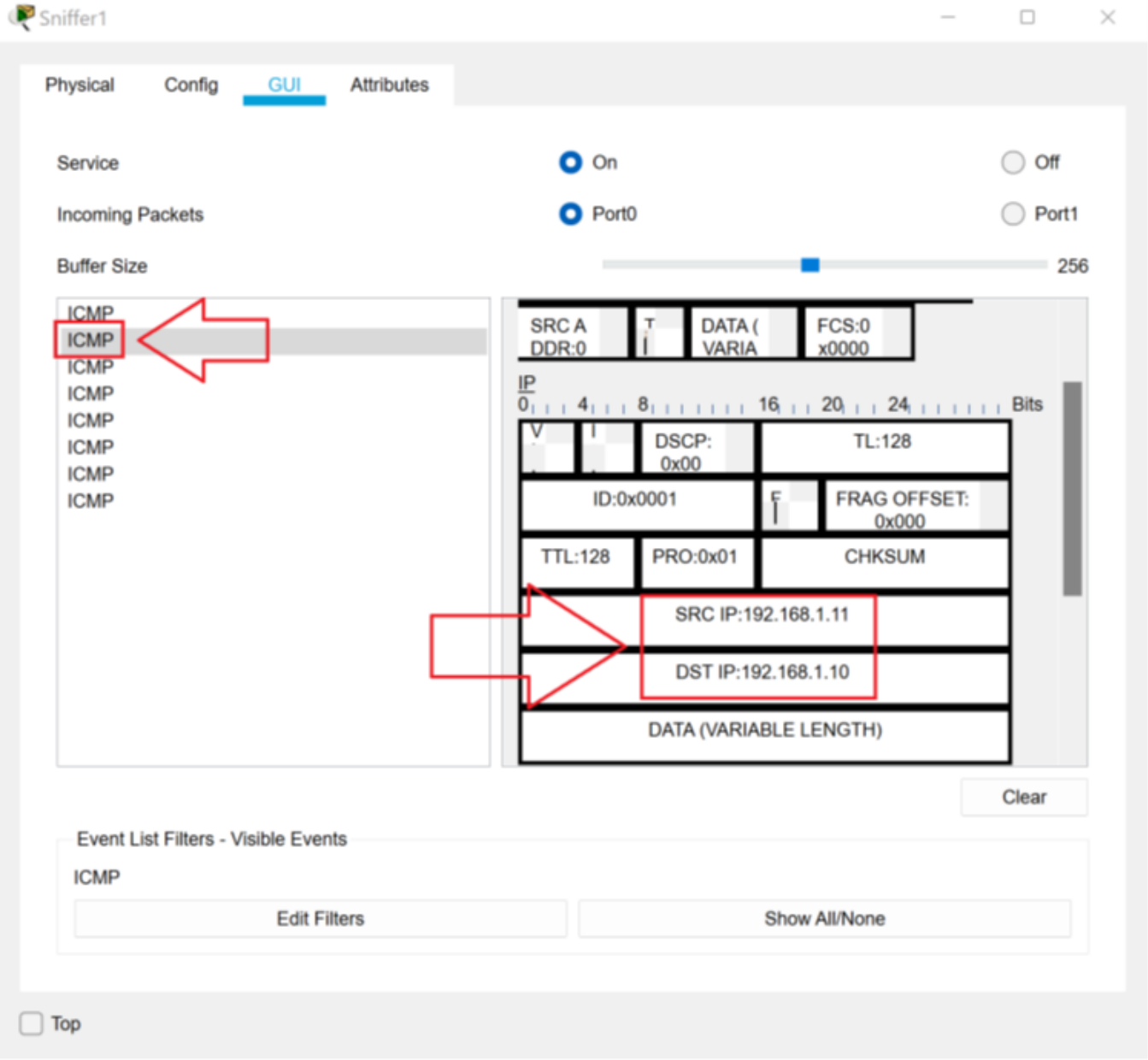
7. In the **Command Prompt**, type **ping 192.168.1.11** and then press **Enter**. You should get four replies.



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.

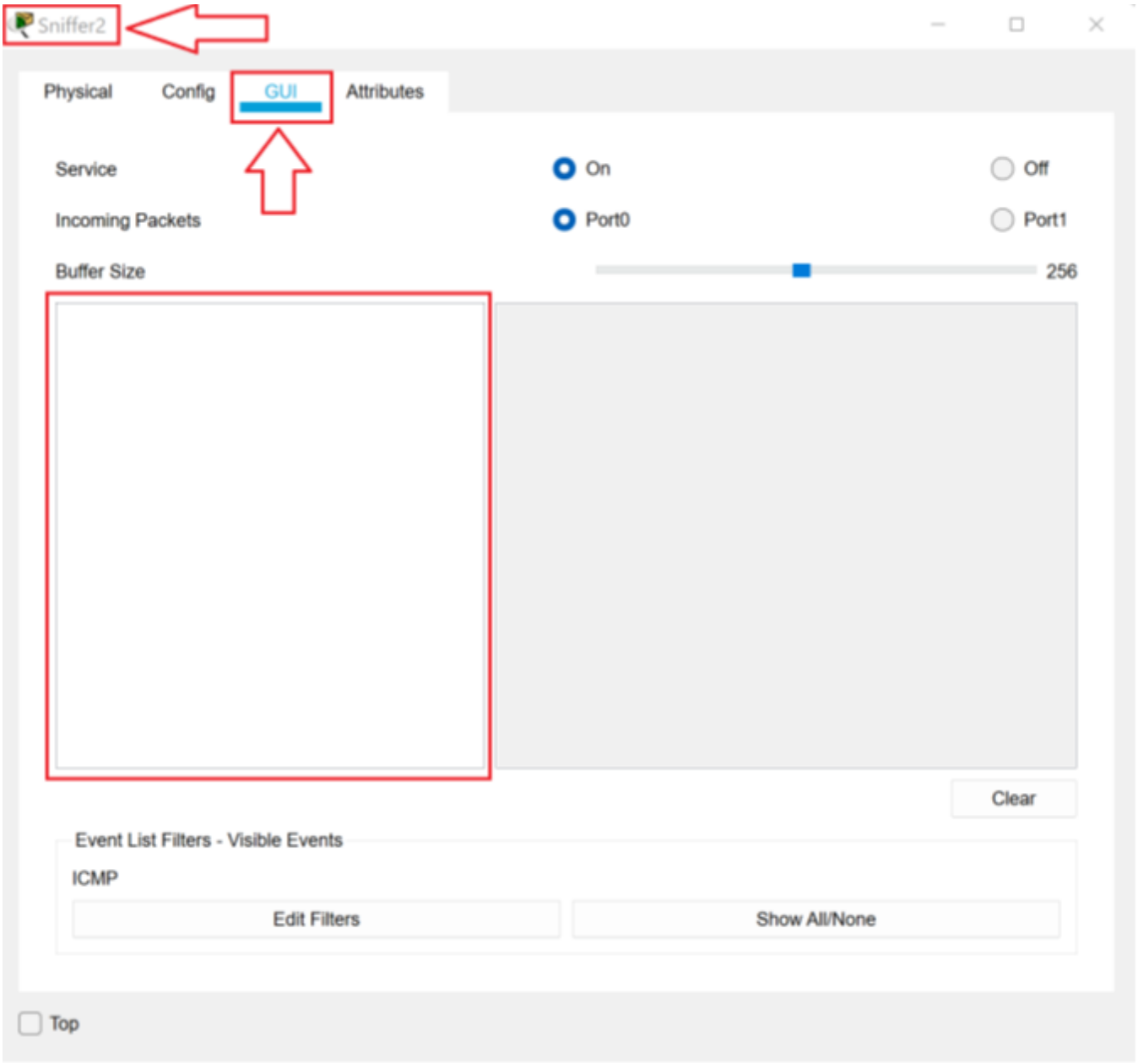


11. The packet sniffer captured the four requests from 192.168.1.10 and the four replies from 192.168.1.11 because the hub sent all of the packets to all of the ports.

TASK B

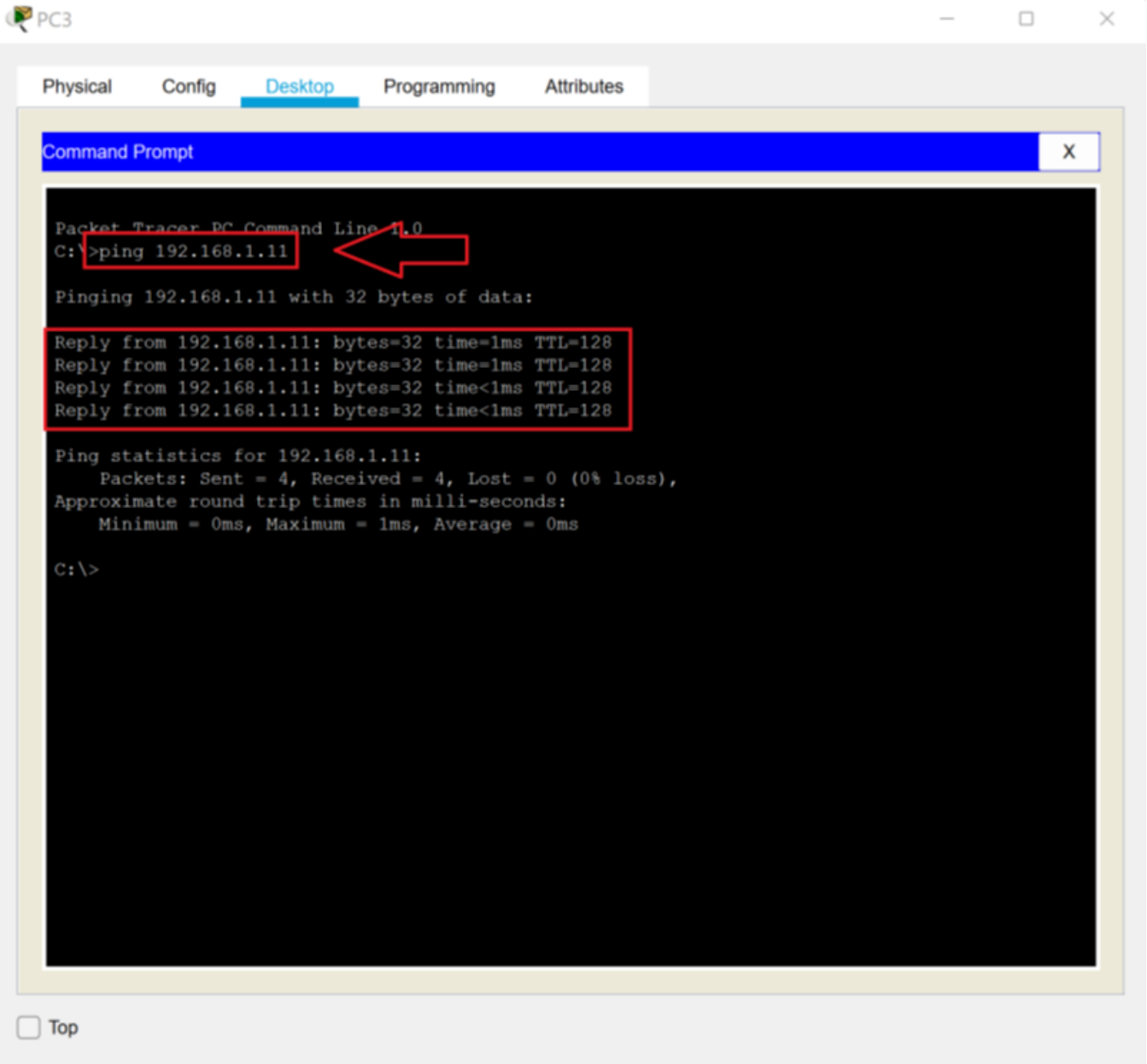
In this task, we will look at the function of a switch.

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.)

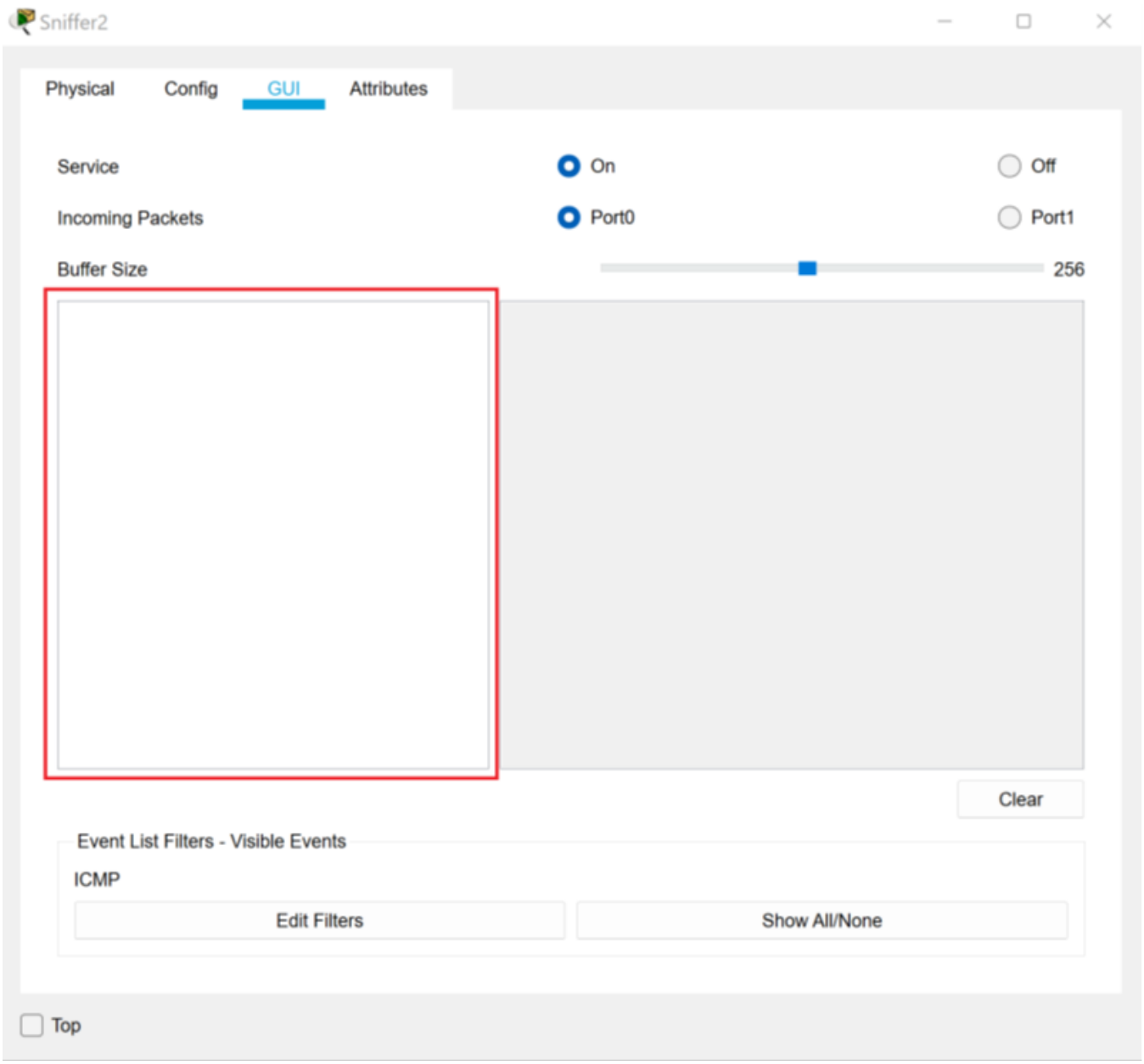


2. Close the **Sniffer2 Properties** dialog box.
3. Click on **PC3** to open the **PC3 Properties** dialog box. Click the **Desktop** tab and then click the **Command Prompt** icon.

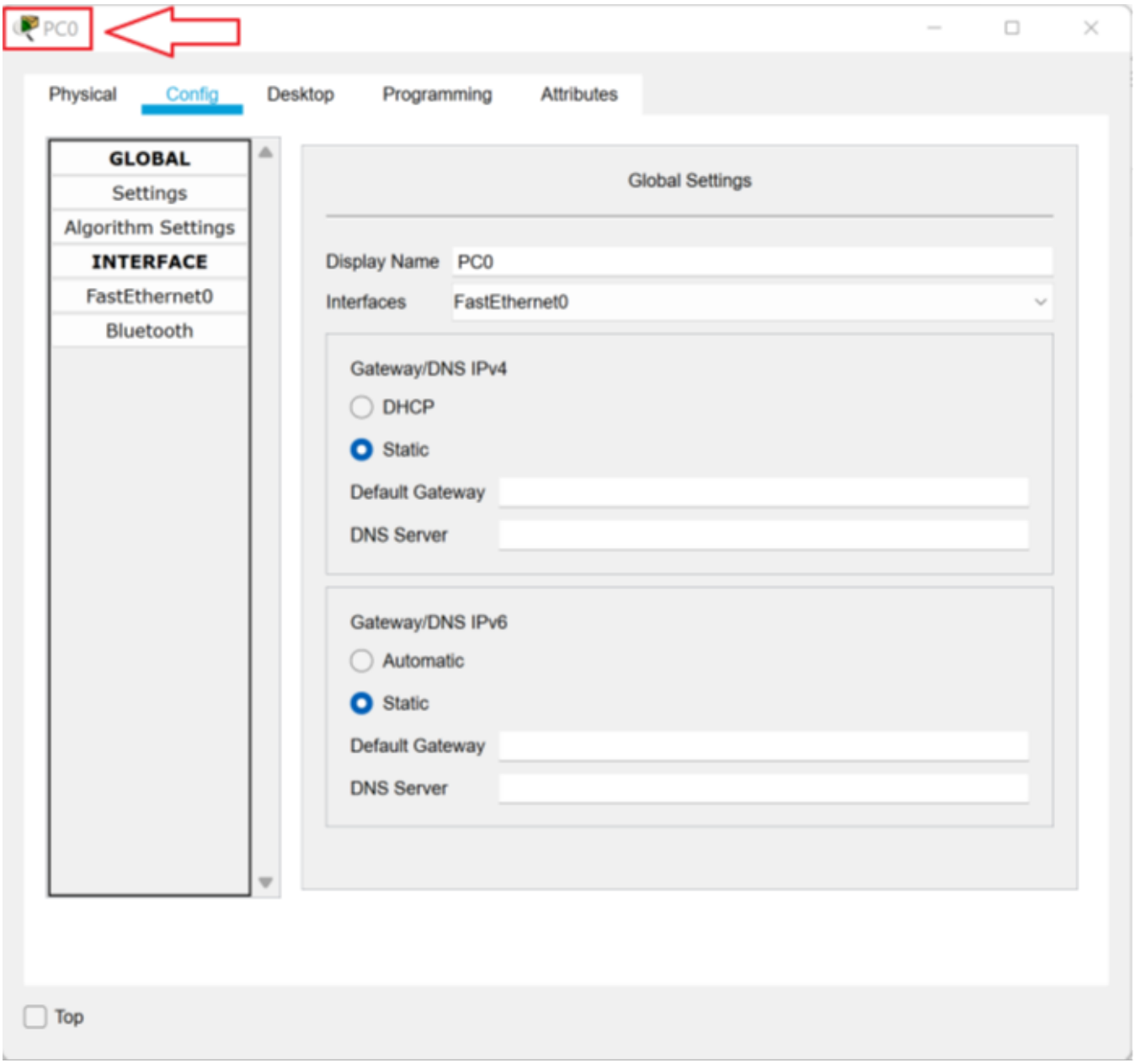
4. In the **Command Prompt**, type **ping 192.168.1.11** and then press **Enter**. You should get four replies.



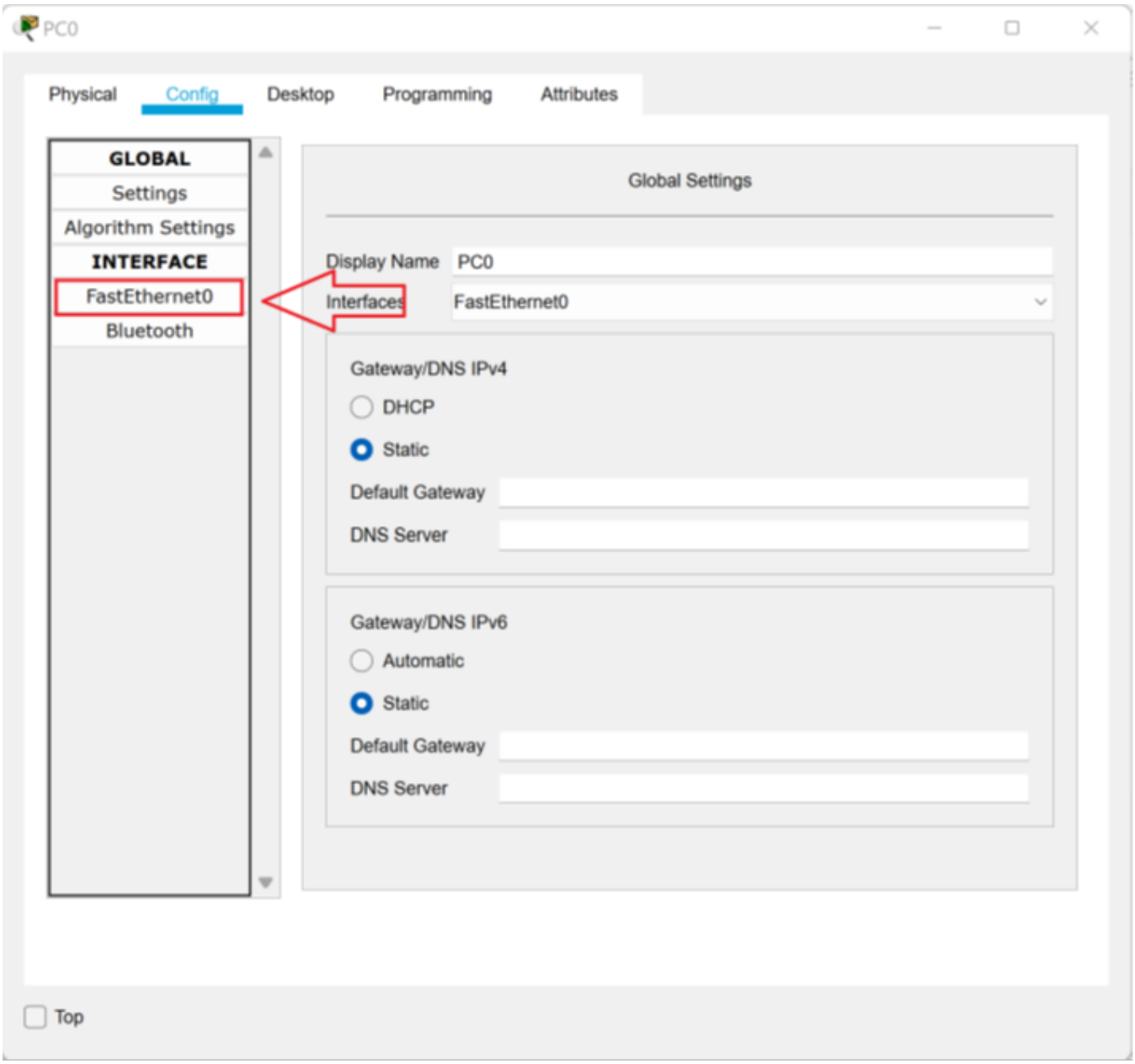
5. Close the **PC3 Properties** dialog box.
6. Click **Sniffer2** to open the **Sniffer2 Properties** dialog box. Notice there are no packets in the buffer. The switch sent the packets directly to the nodes being addressed. Since none of the packets were addressed to the MAC address of Sniffer2, none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



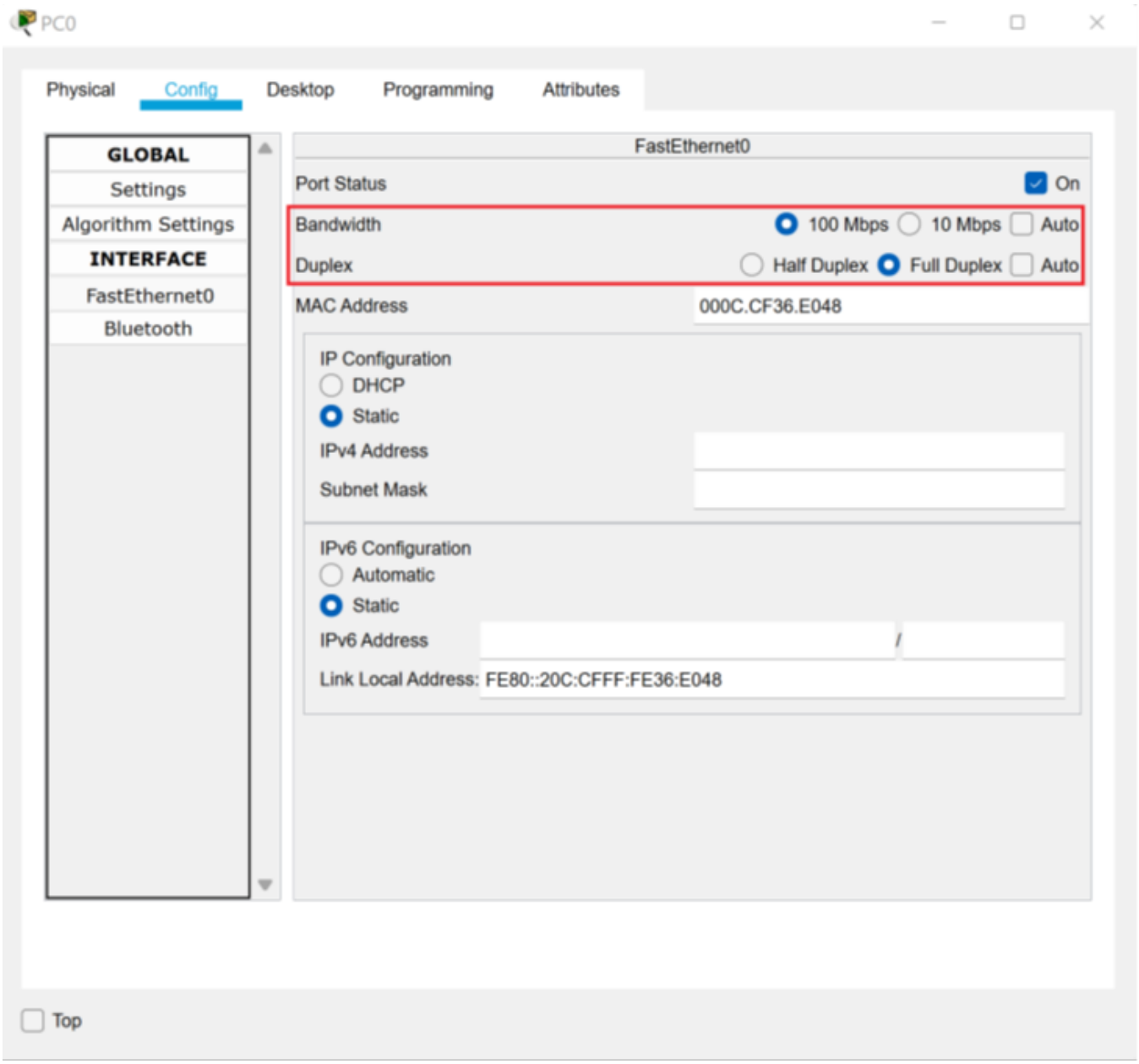
7. Close the **3.4.2 Lab File** file.



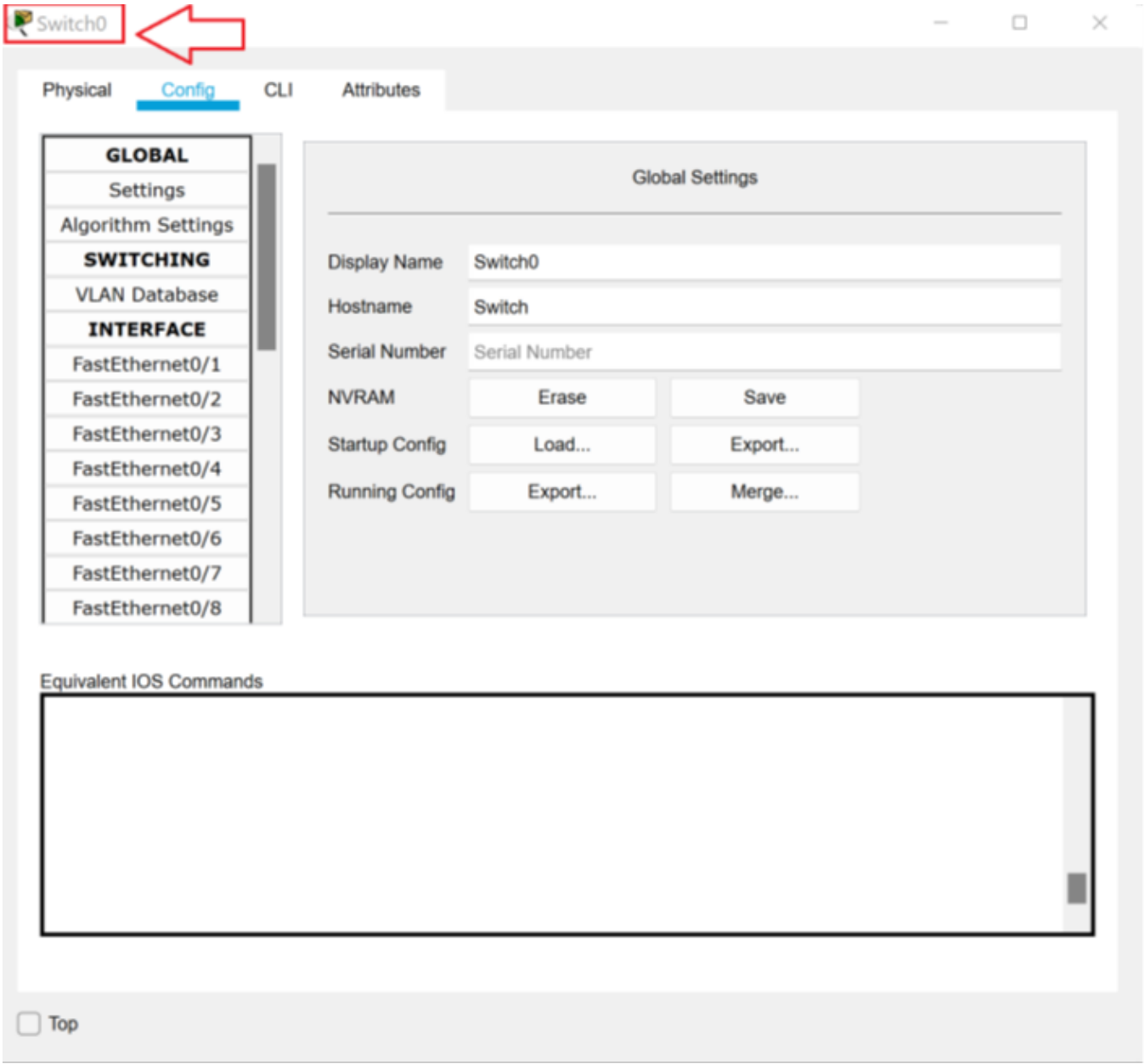
4. In the **Interface** menu, click **FastEthernet0**.



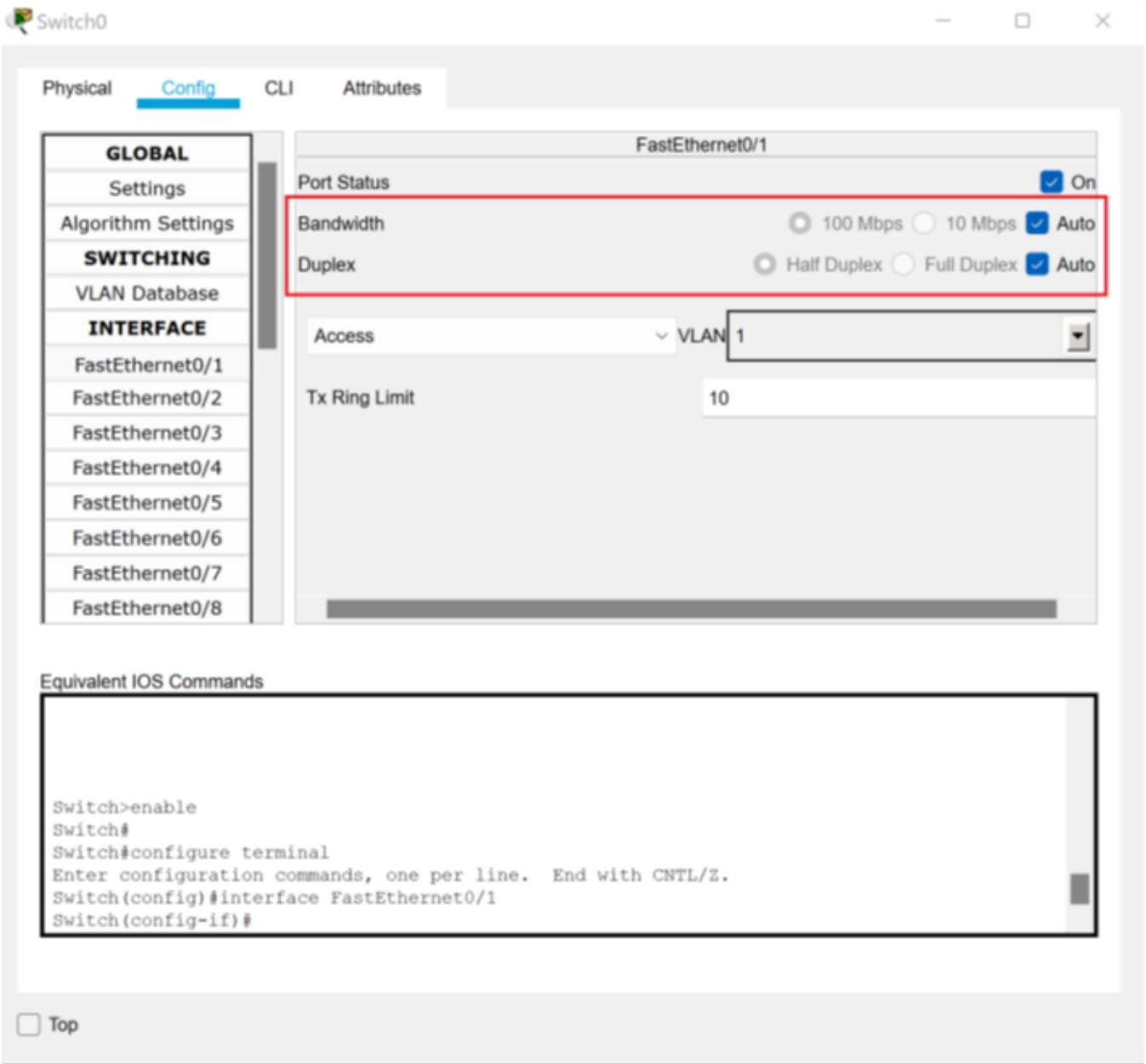
5. Notice the network card is set to **100 Mbps Full Duplex**.



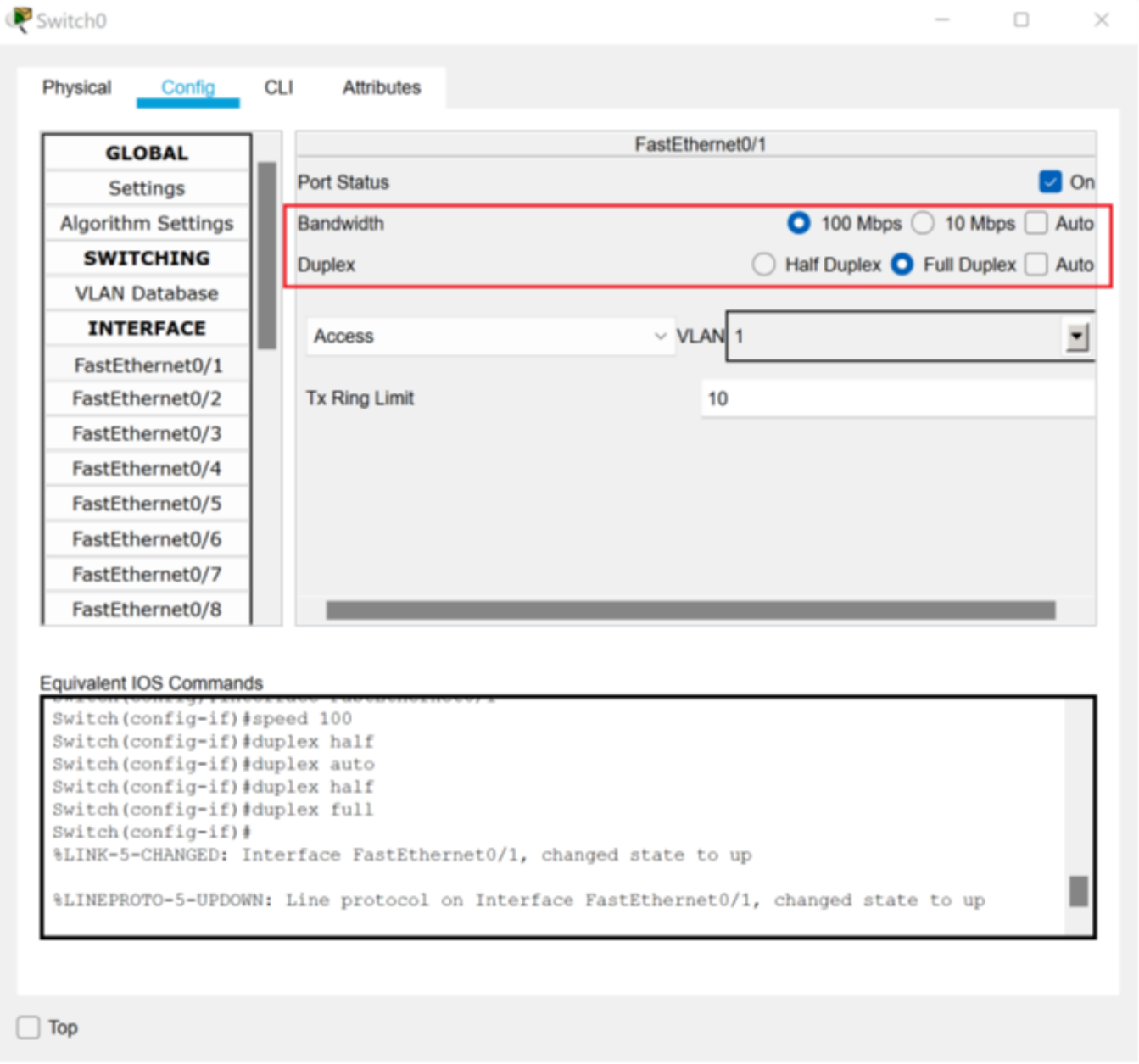
6. Close the **PC0 Properties** dialog box.
7. Click on **Switch0** to open the **Switch0 Properties** dialog box.



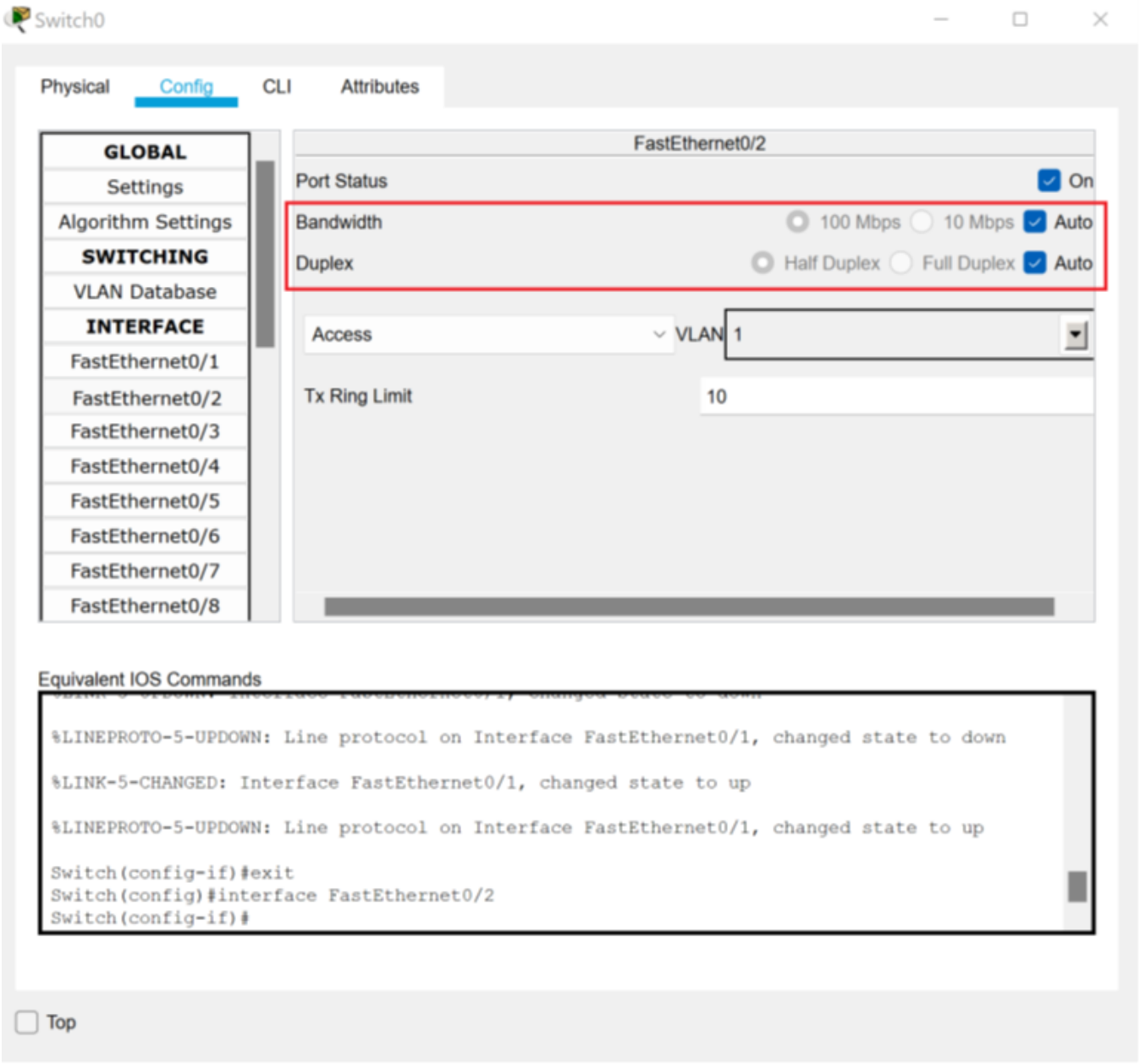
8. In the **Switch0 Properties** dialog box, in the **Interface** menu, click **FastEthernet0/1**. (This is the port on the switch to which the PC0 cable is connected.) Observe that the settings do not match the settings on the client.



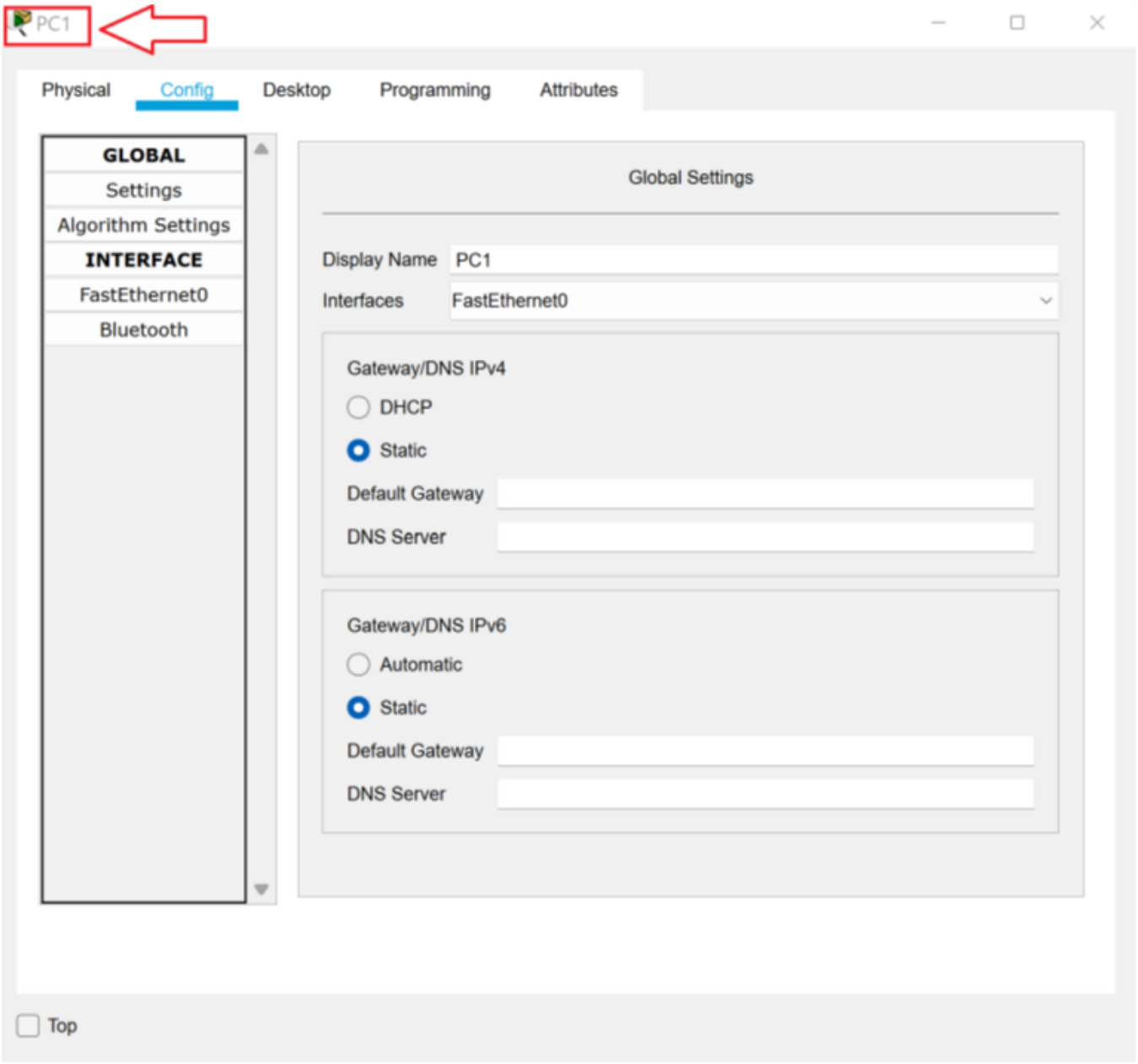
9. To fix the problem, uncheck the **Auto bandwidth** check box. Then uncheck the **Auto Duplex** check box. Select the **Full Duplex** radio button so that the settings on the switch match the settings on the client.



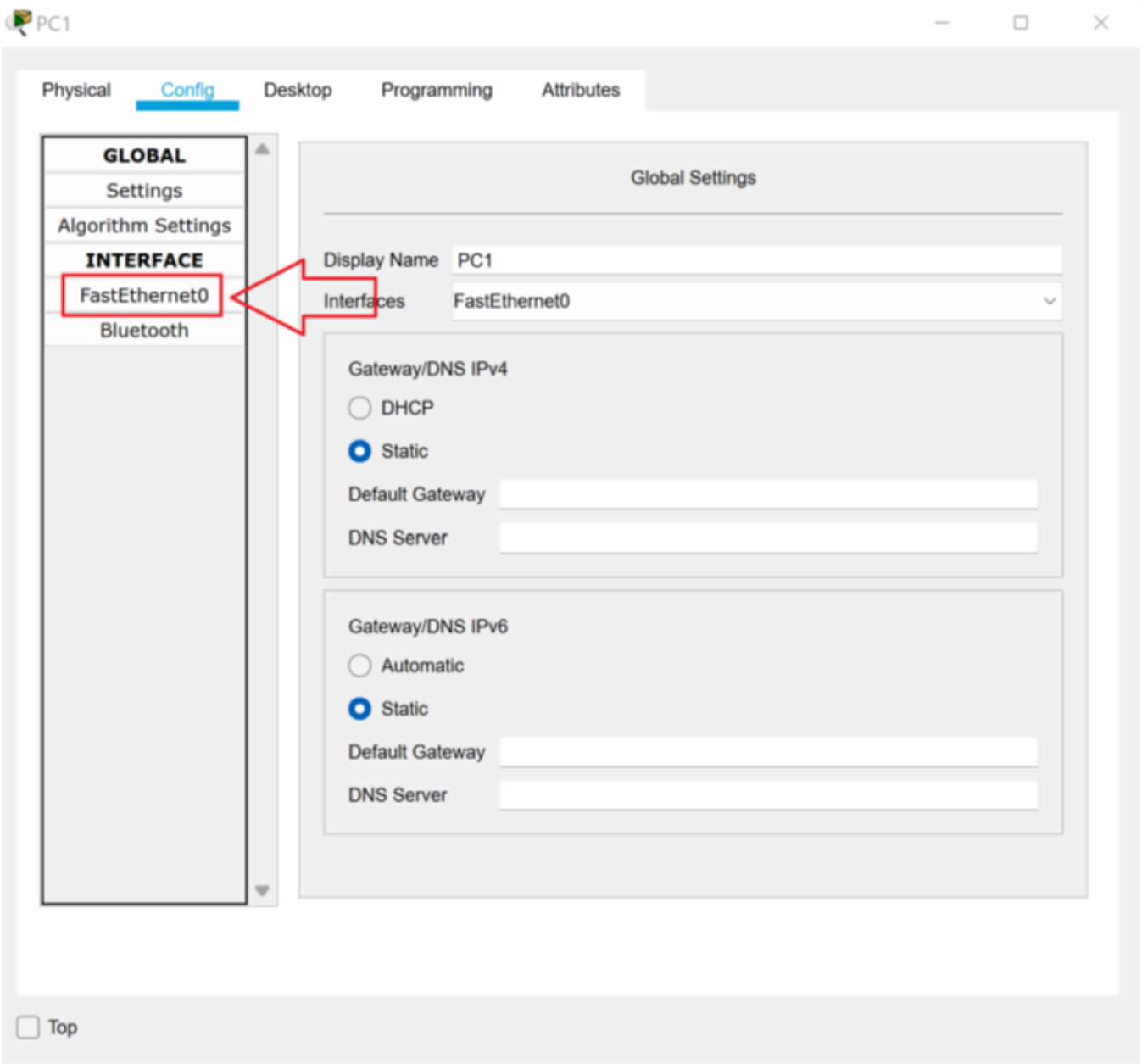
10. Close the **Switch0Properties** dialog box.
11. Notice that the connection between **PC0** and **Switch0** has been resolved.
12. Click **Switch0** to open the **Switch0 Properties** dialog box.
13. In the Interface menu, click **FastEthernet0/2**. (This is the port on the switch to which the PC1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



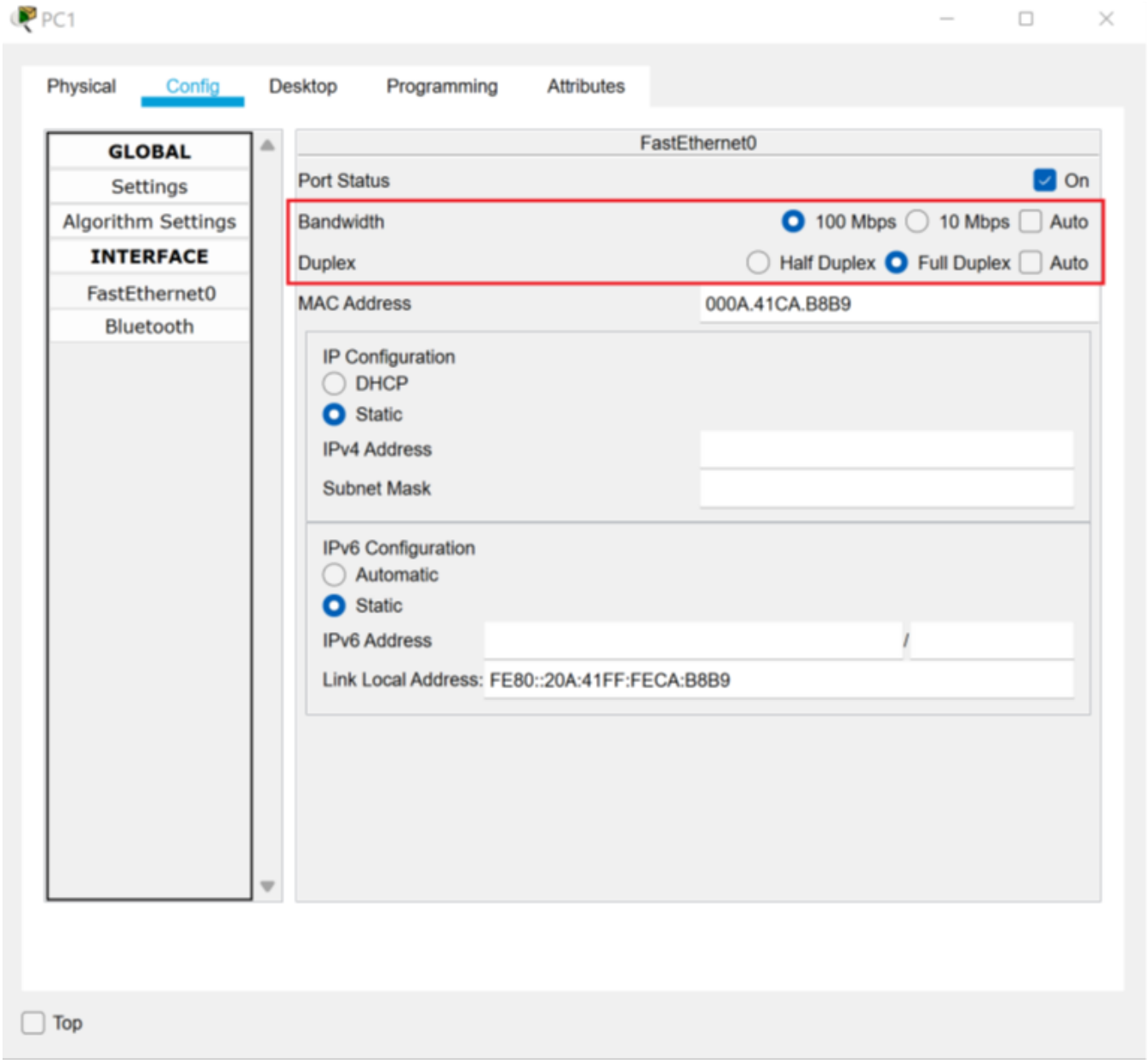
14. Close the **Switch0 Properties** dialog box.
15. Click on **PC1** to open the **PC1 Properties** dialog box.



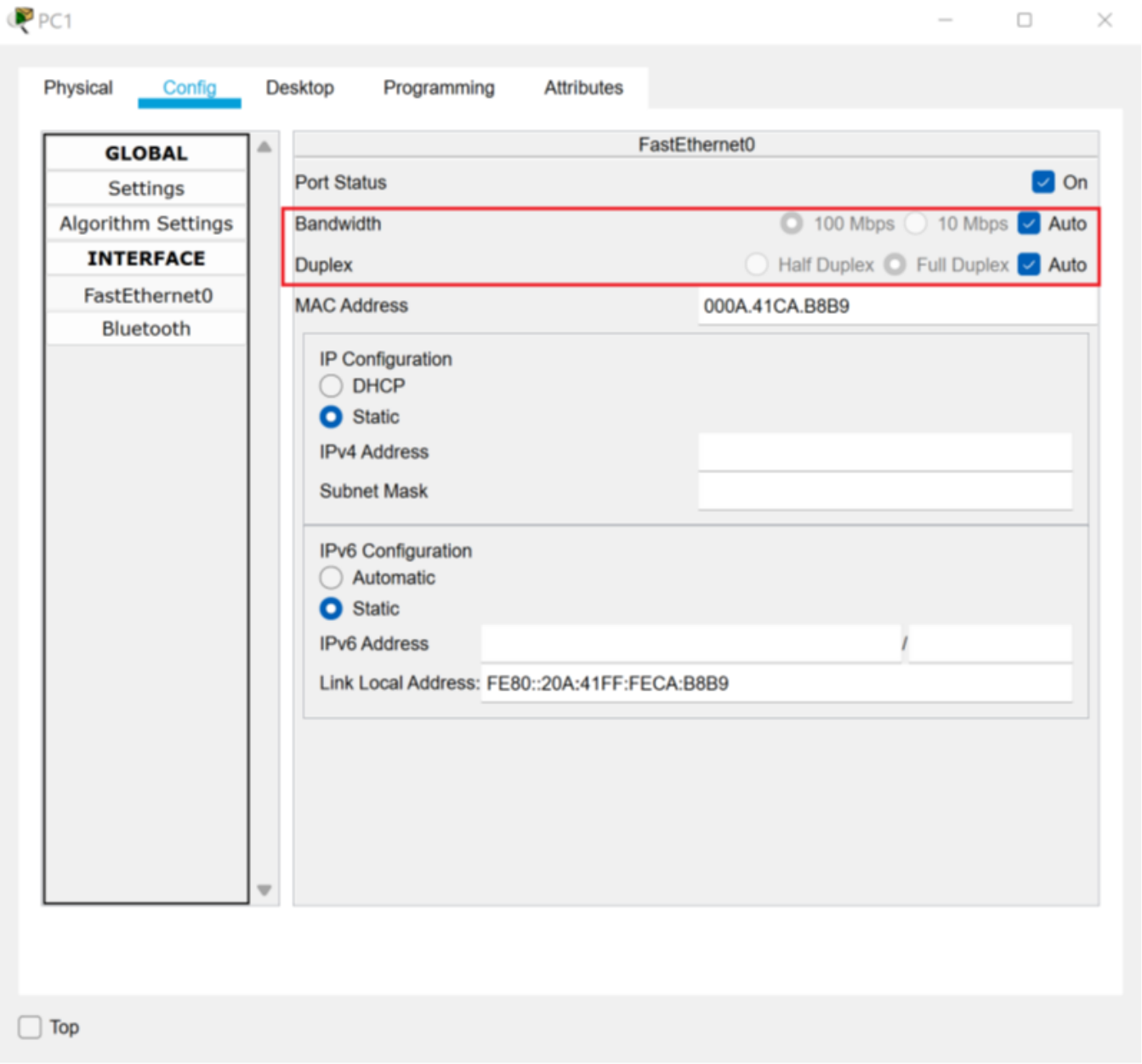
16. In the **Interface** menu, click **FastEthernet0**.



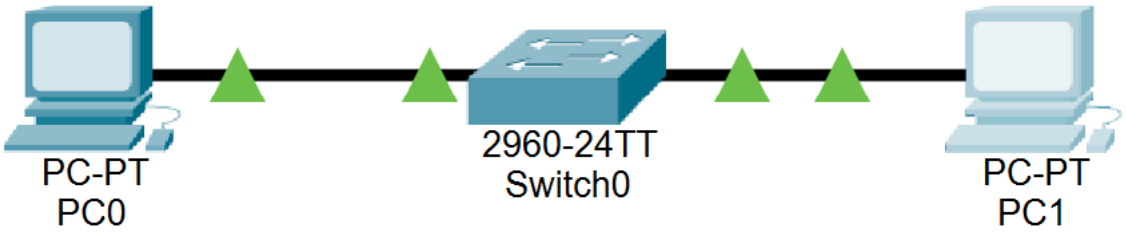
17. Notice the network card is set to **100 Mbps Full Duplex**.



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.

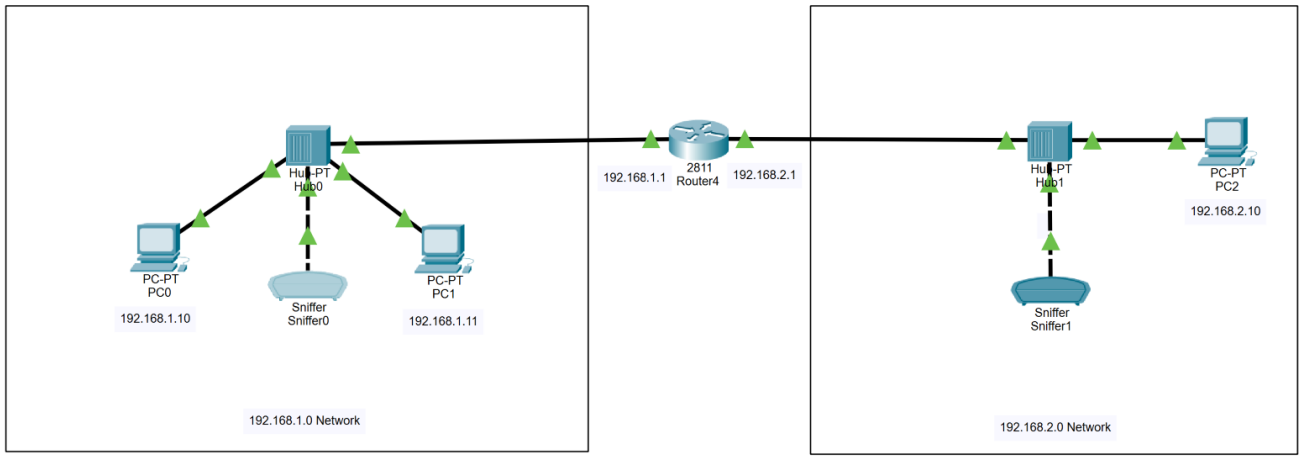
TASK A

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

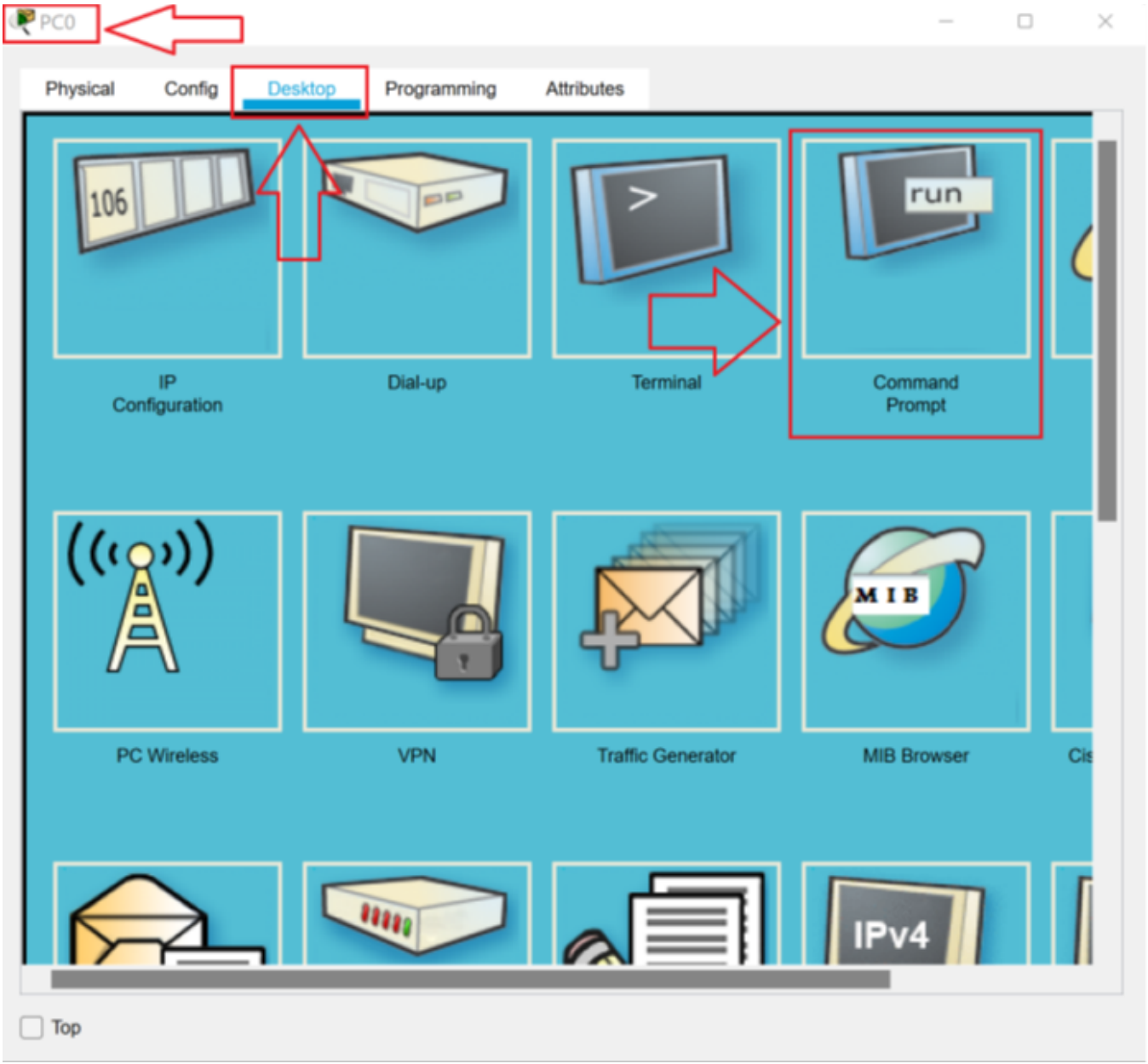
1. Download the **3.4.3 Lab File** and open it in **Packet Tracer**.

3.4.3 Lab File

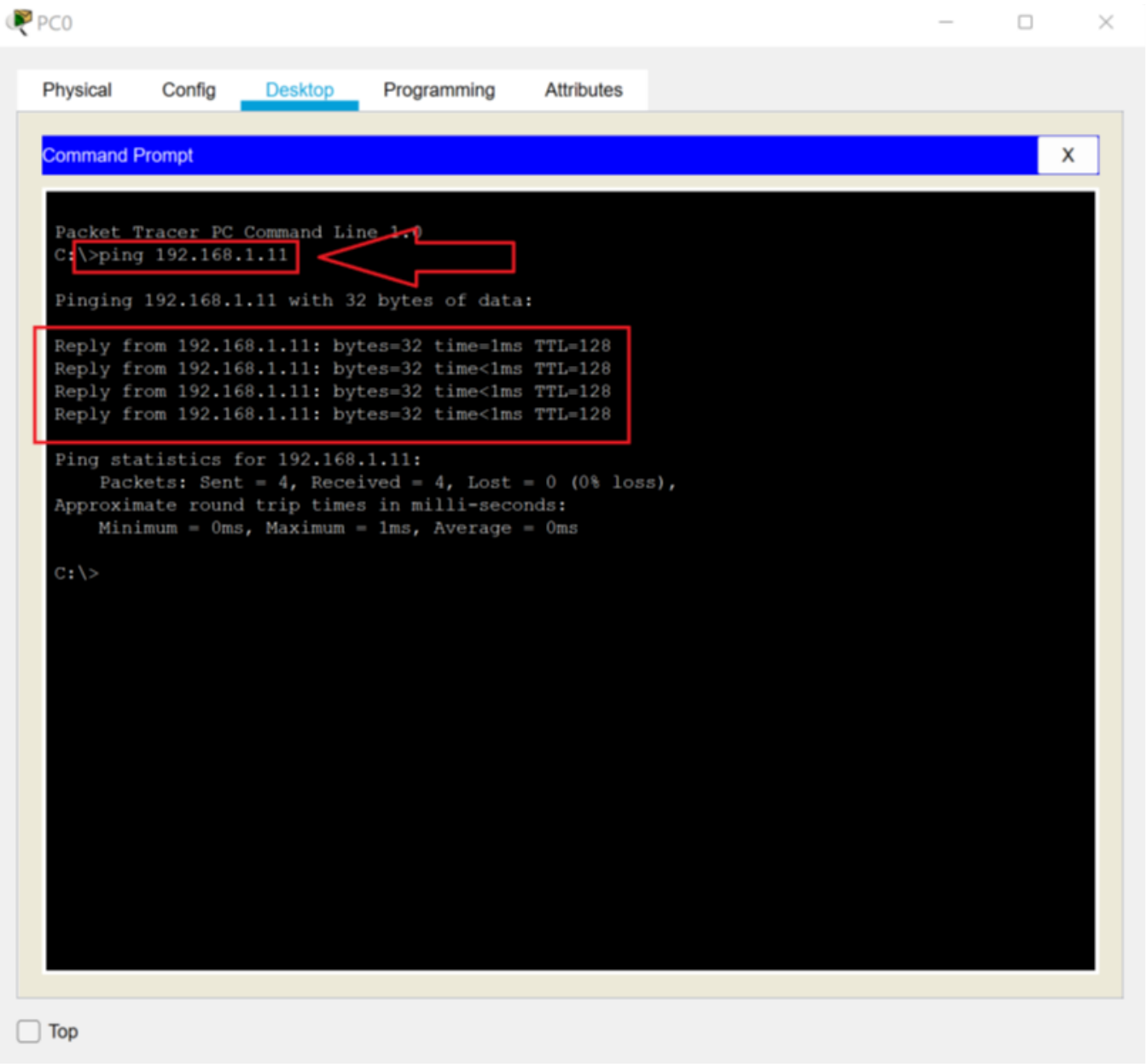
PKT File



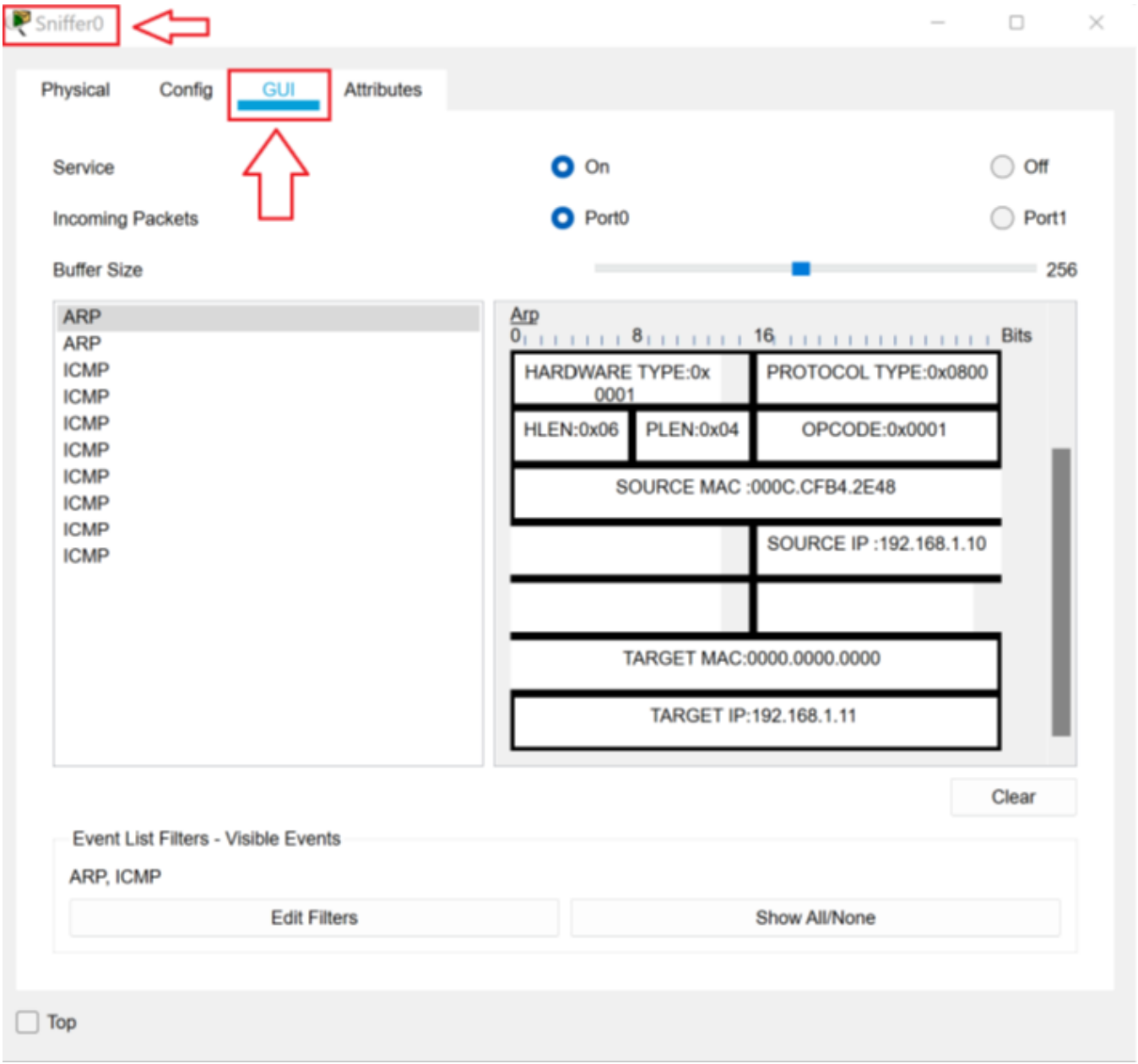
2. Click on **PC0** to open the **PC0 Properties** dialog box. Select the **Desktop** tab, and then click the **Command Prompt** icon.



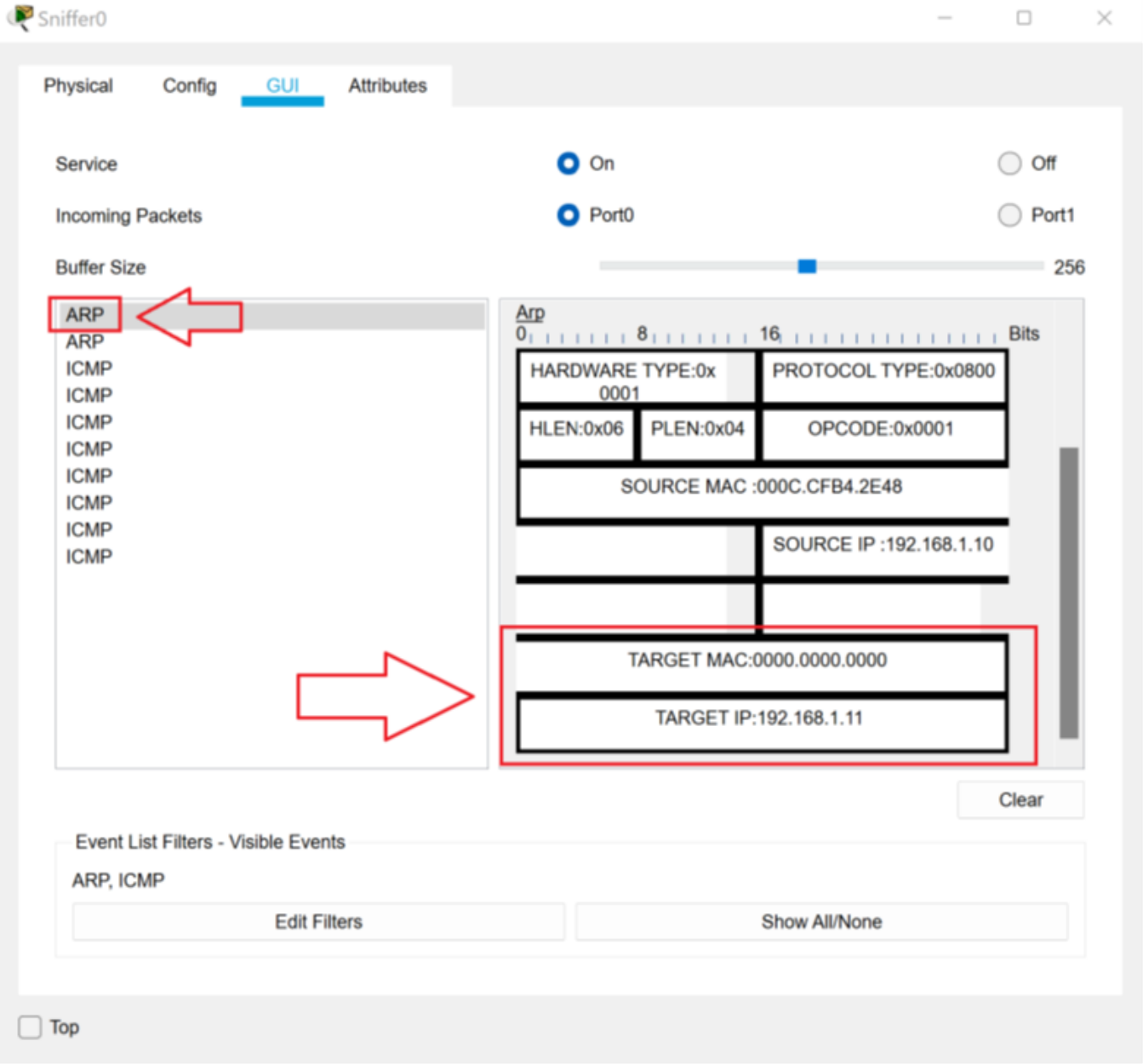
3. In the **Command Prompt**, type **ping 192.168.1.11** and then press **Enter**. You should get four replies.



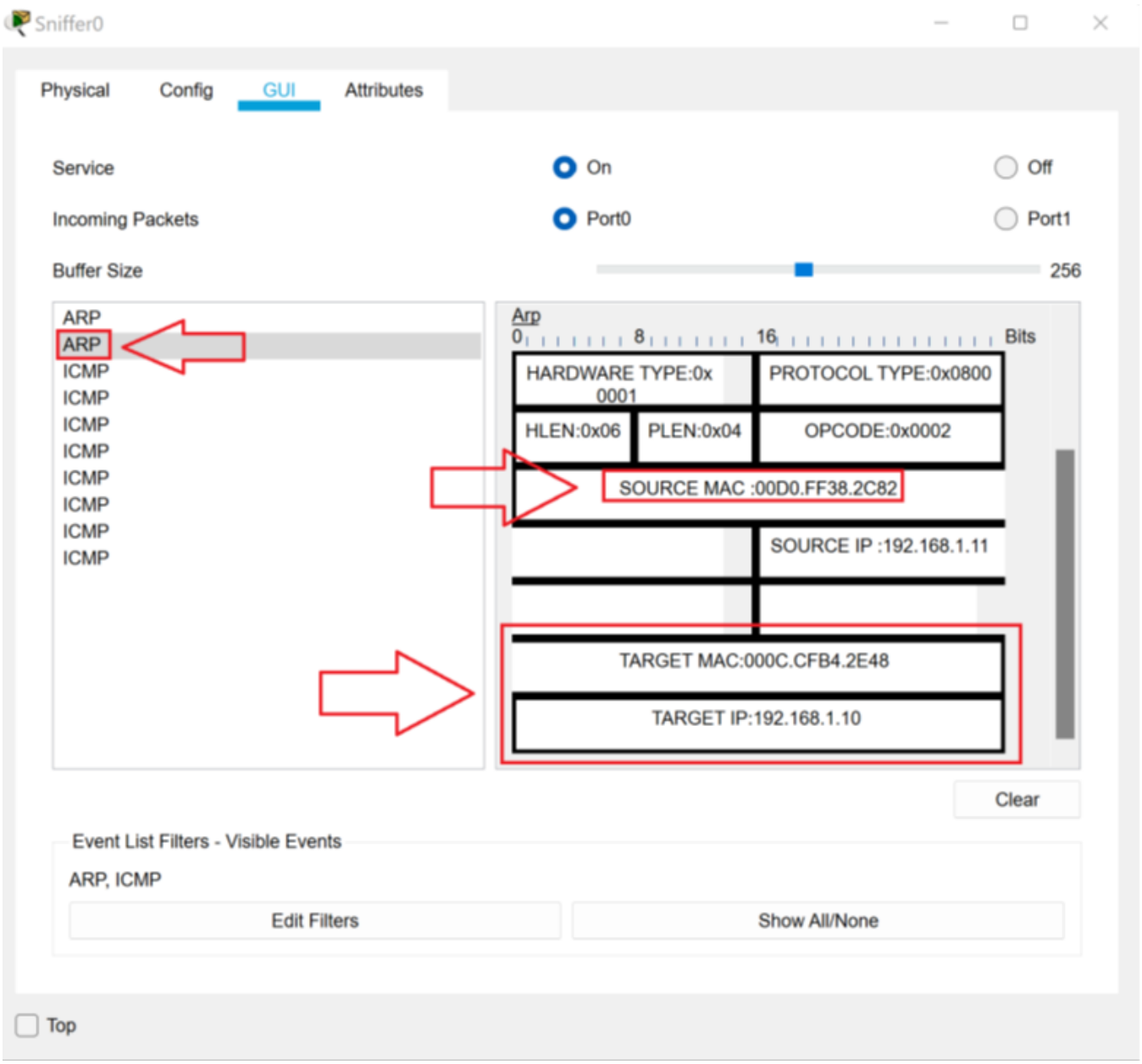
4. Close the **PC0 Properties** dialog box.
5. Click **Sniffer0** to open the **Sniffer0 Properties** dialog box. Then click the **GUI** tab.



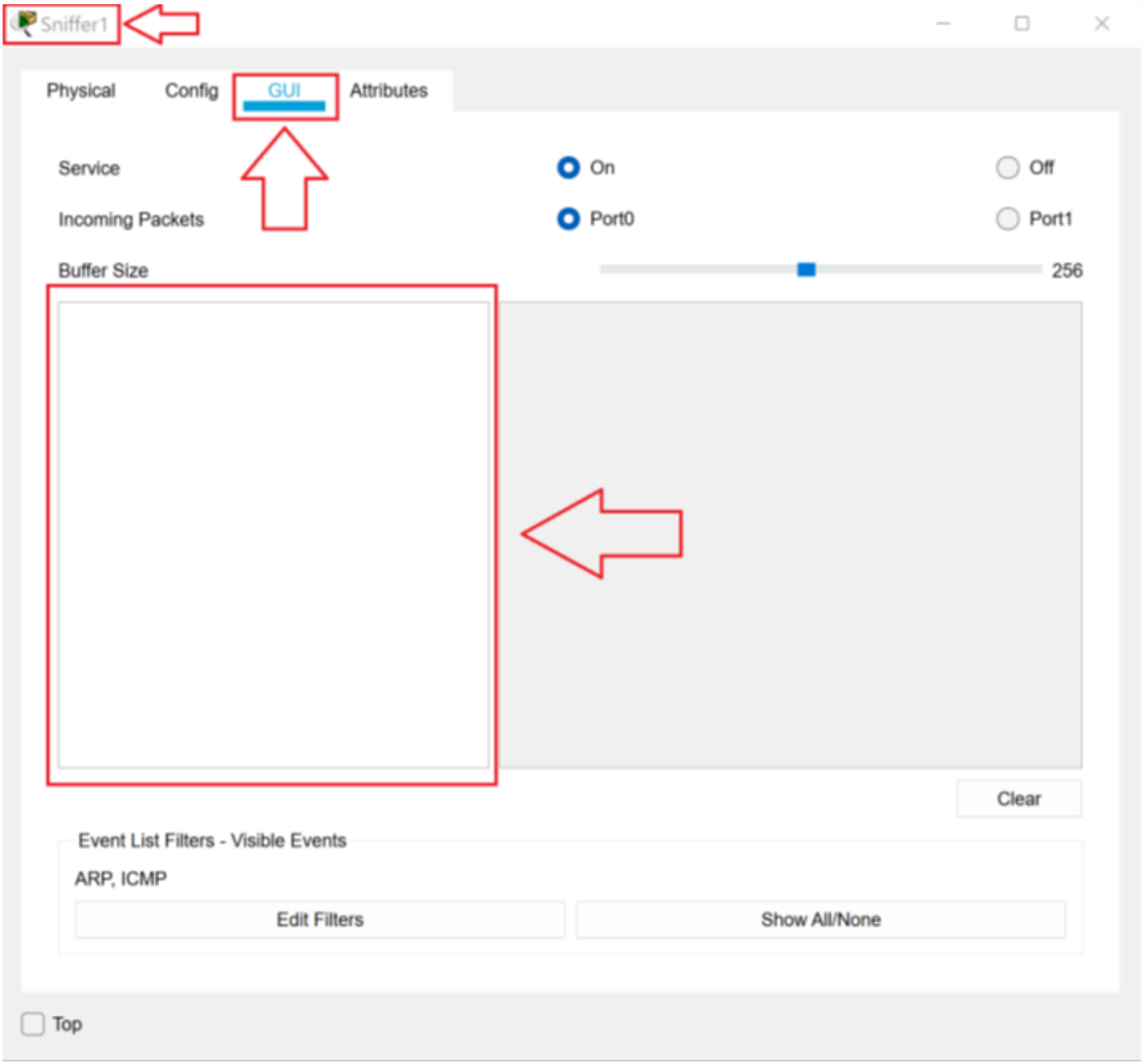
6. You should see two ARP packets and eight ICMP packets. (Don't worry if there are any other packets showing or if there are less than eight ICMP packets. If you have both ARP packets and some of the ICMP packets everything is fine.) Click on the first **ARP** packet. Scroll down in the ARP packet until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

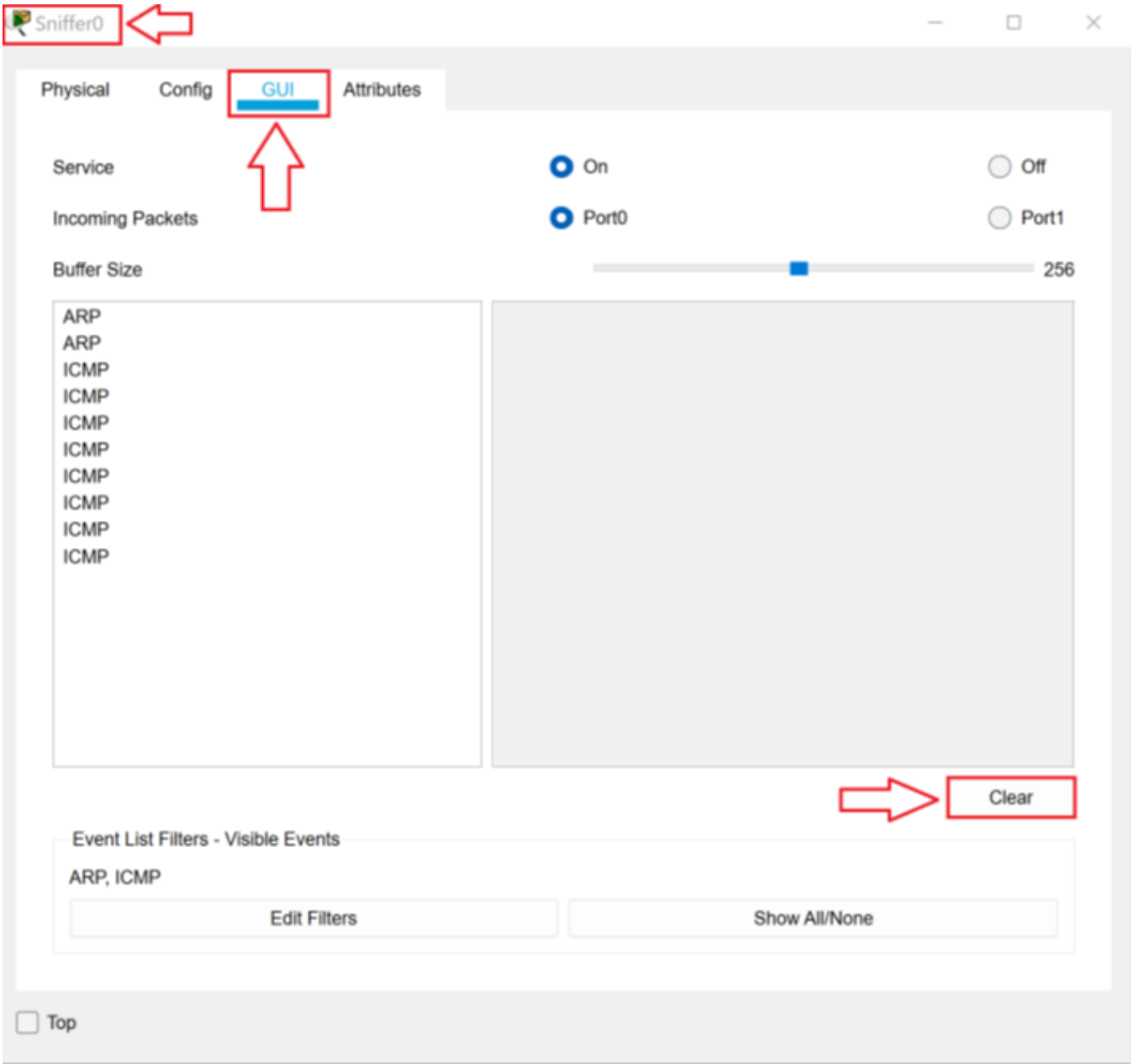


12. Close the **Sniffer1 Properties** dialog box.

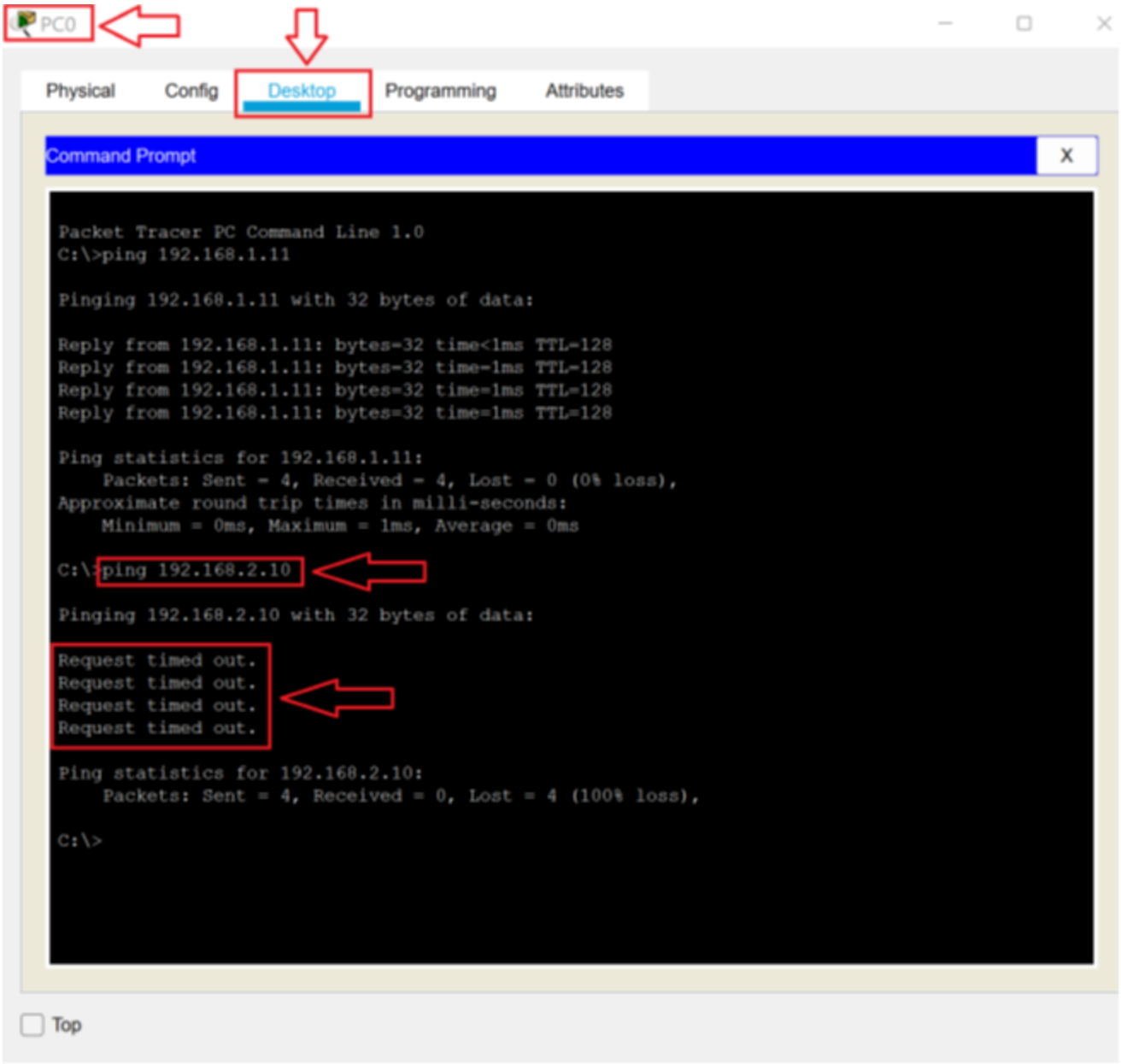
TASK B

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

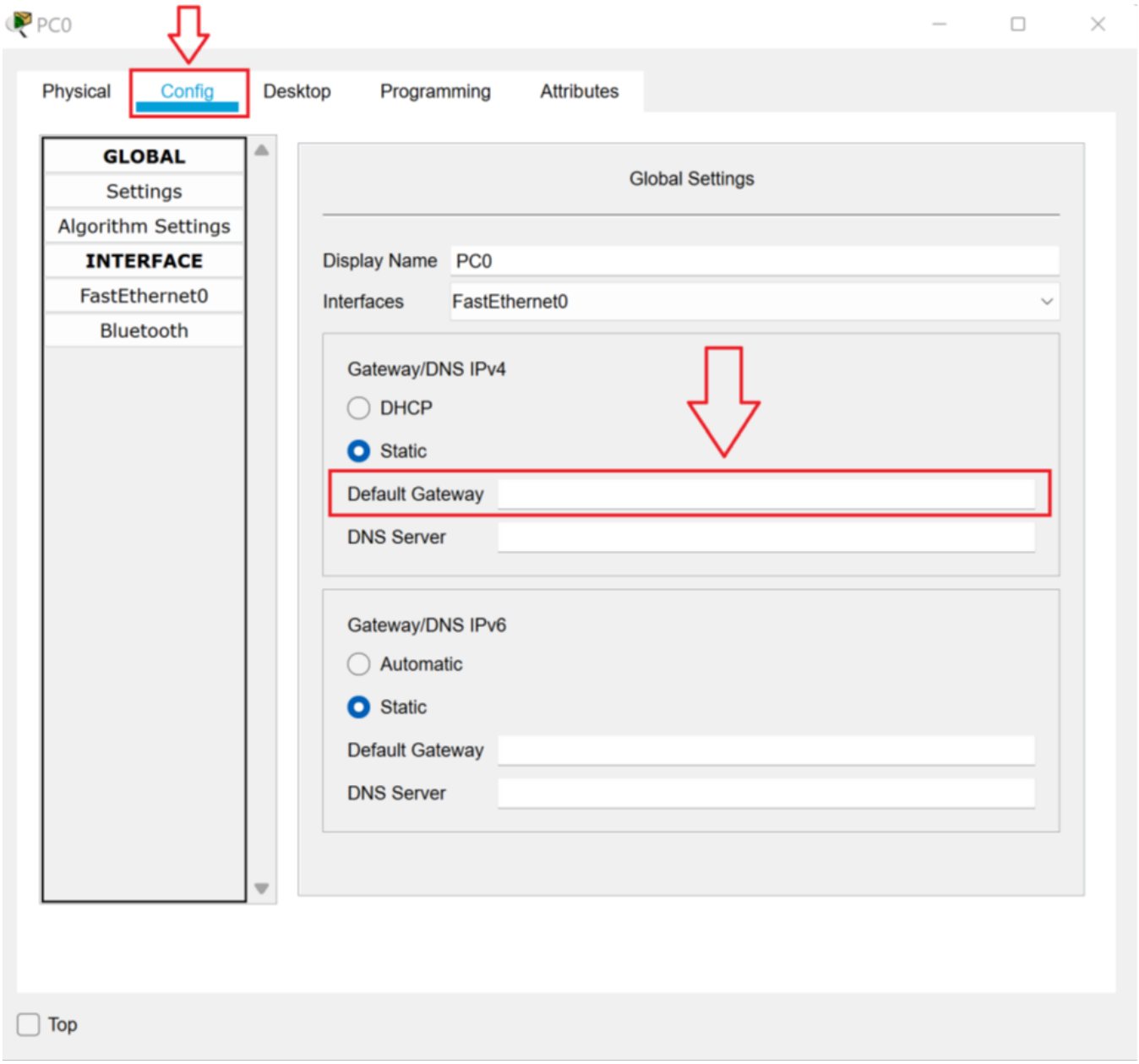
1. Click **Sniffer0**. On the **GUI** tab, click the **Clear** button to clear the buffer.



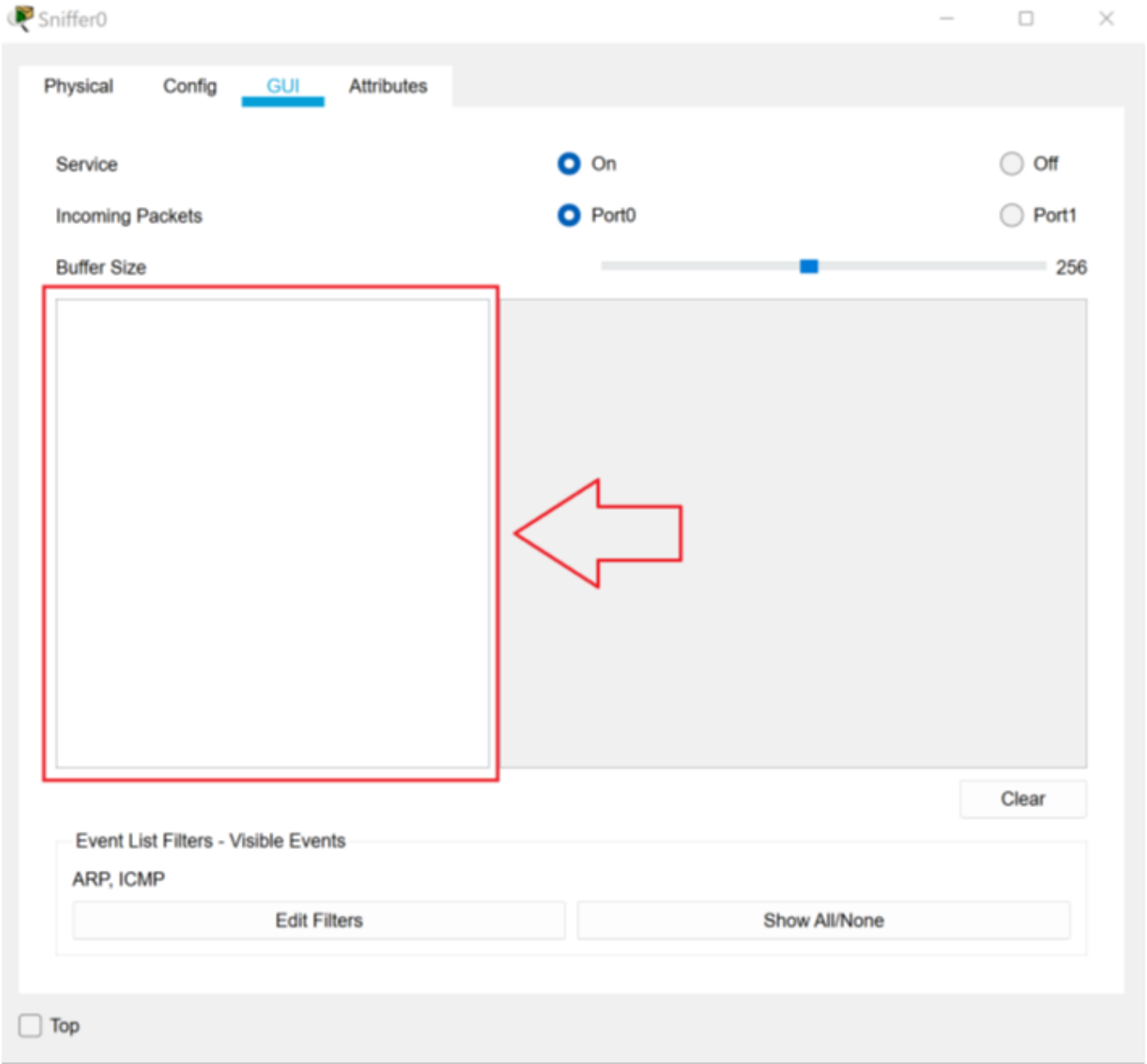
2. Close the **Sniffer0 Properties** dialog box.
3. Click **PC0**. In the **Desktop** tab, in the **Command Prompt**, type **ping 192.168.2.10** and then press Enter. Notice that 192.168.2.10 does not respond.



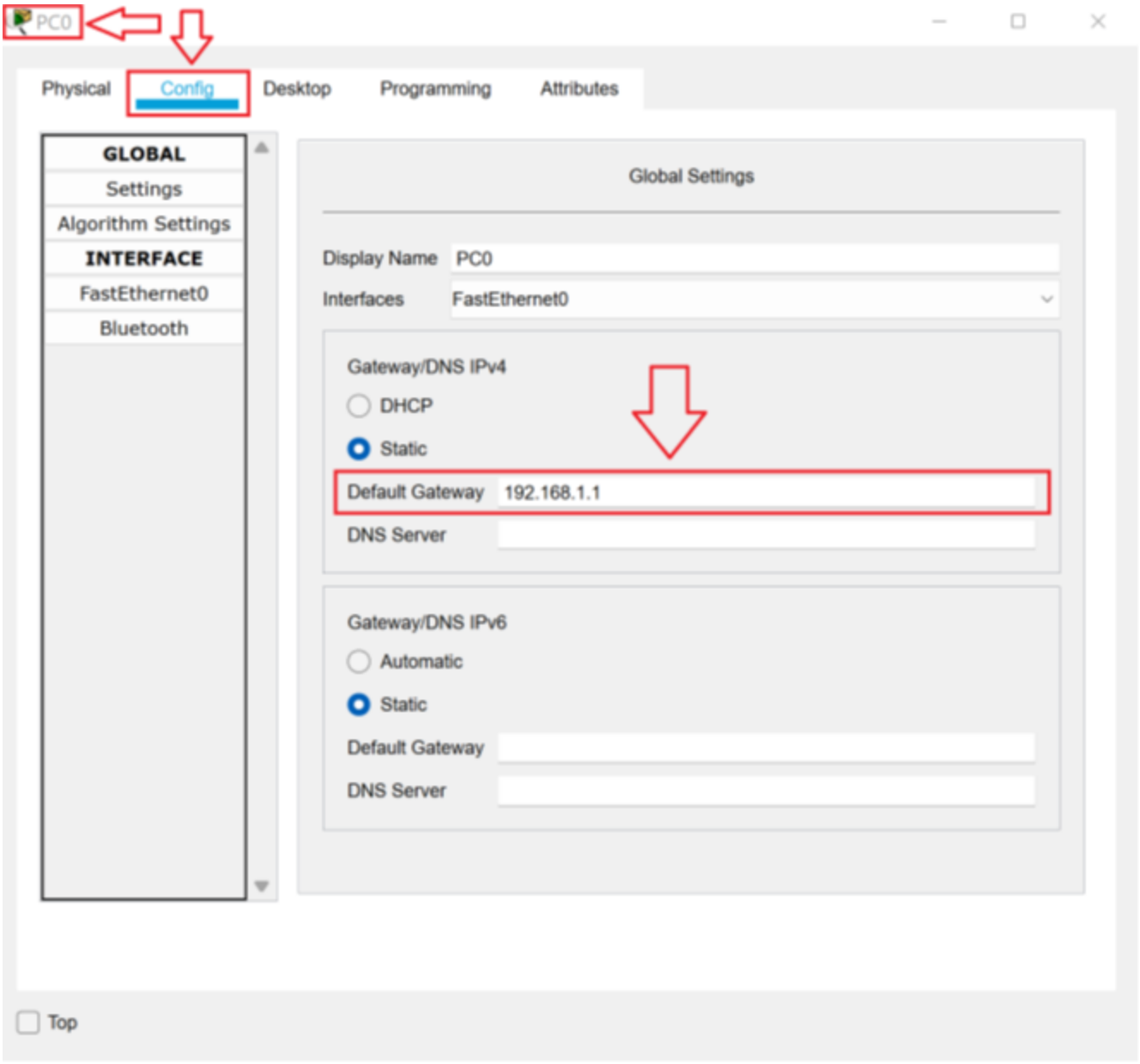
4. In the **PC0 Properties** dialog box, click the **Config** tab. Notice that PC0 does not have a default gateway setting. Since PC0 does not know the address of the router, it cannot send any packets to remote networks.



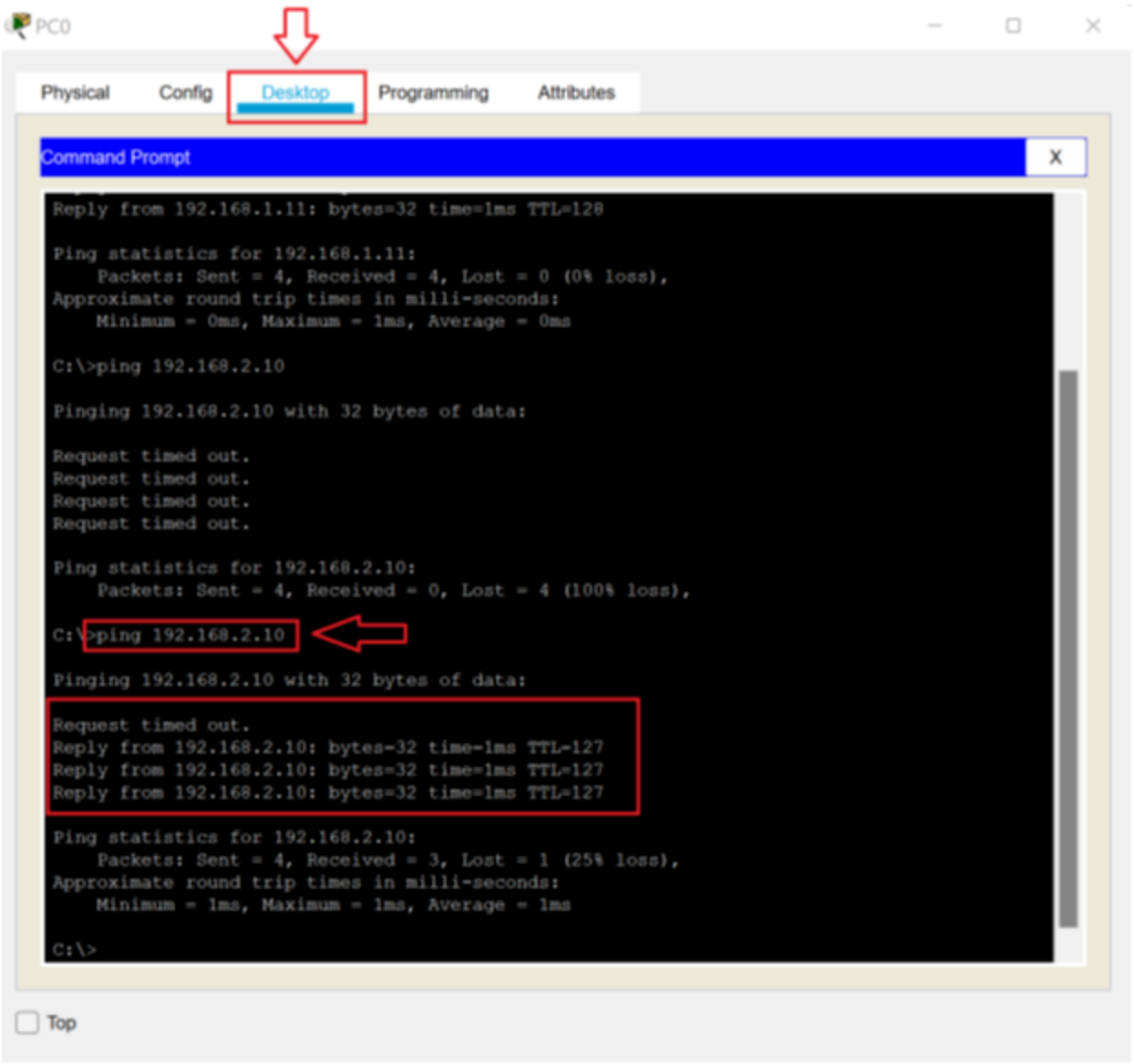
5. Close the **PC0 Properties** dialog box.
6. Click **Sniffer0**. Notice that Sniffer0 does not have any packets in its buffer. Because PC0 knows that the traffic was destined for a different network, it did not send out any ARP broadcasts for 192.168.2.10.



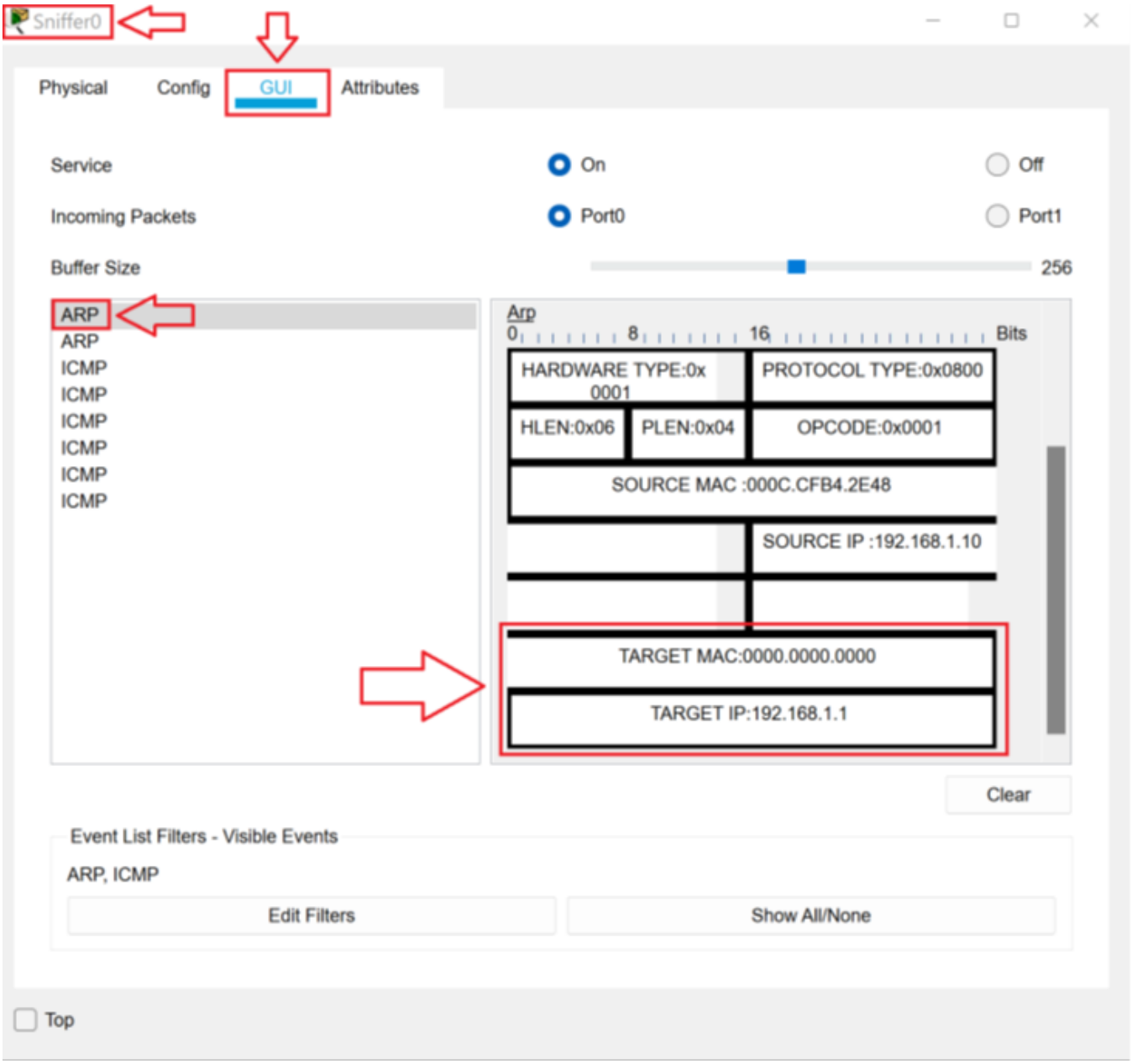
7. Close the **Sniffer0 Properties** dialog box.
8. Click **PC0**. On the **Config** tab, in the **Default Gateway** text box, type **192.168.1.1**.



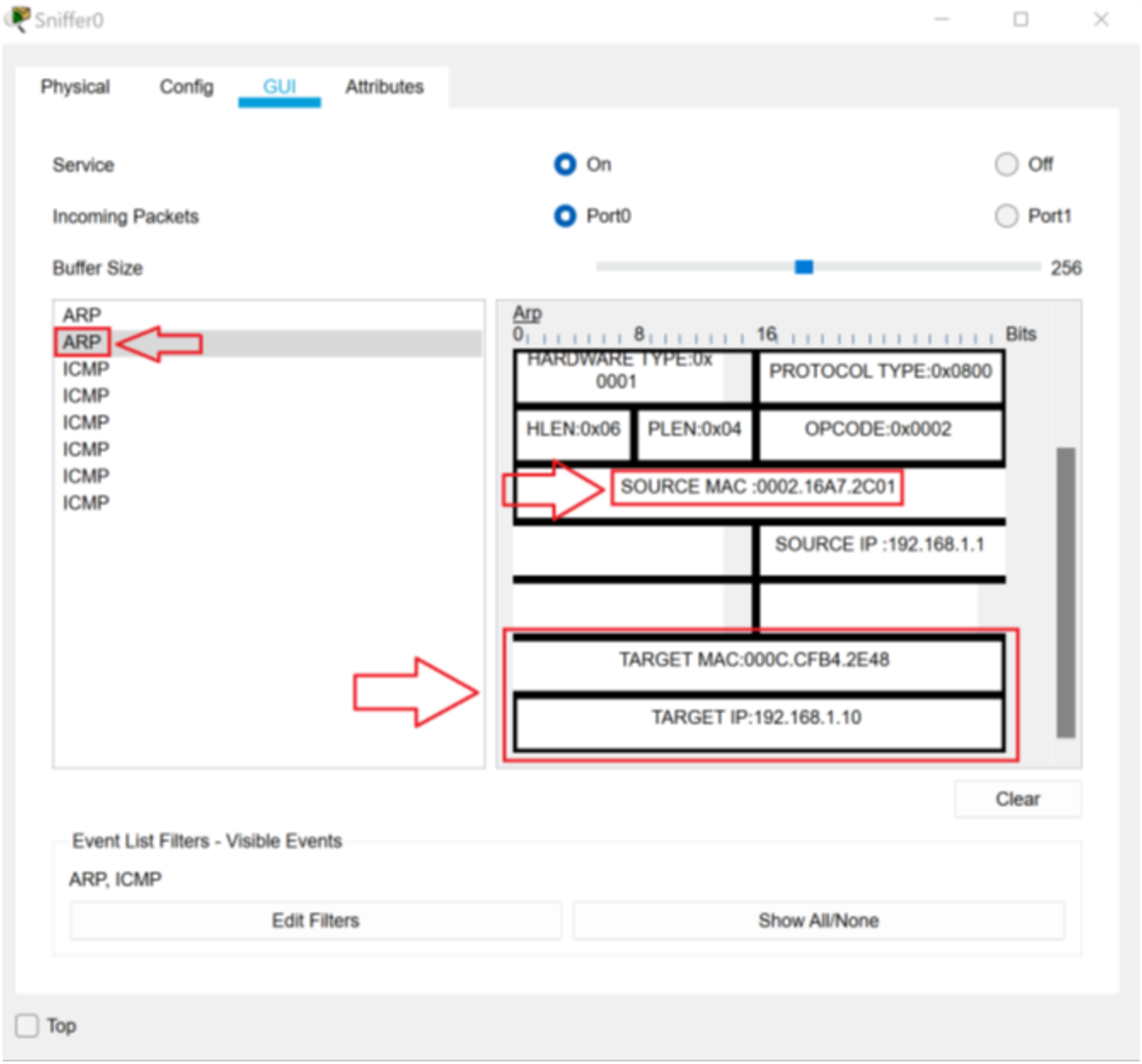
9. Click the **Desktop** tab. In the **Command Prompt**, press the **up-arrow** key on your keyboard to recall the last command (ping 192.168.2.10) and then press **Enter**. Notice that you now get a reply from 192.168.2.10. (Do not be concerned if the first one or two replies time out. It may take PC0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



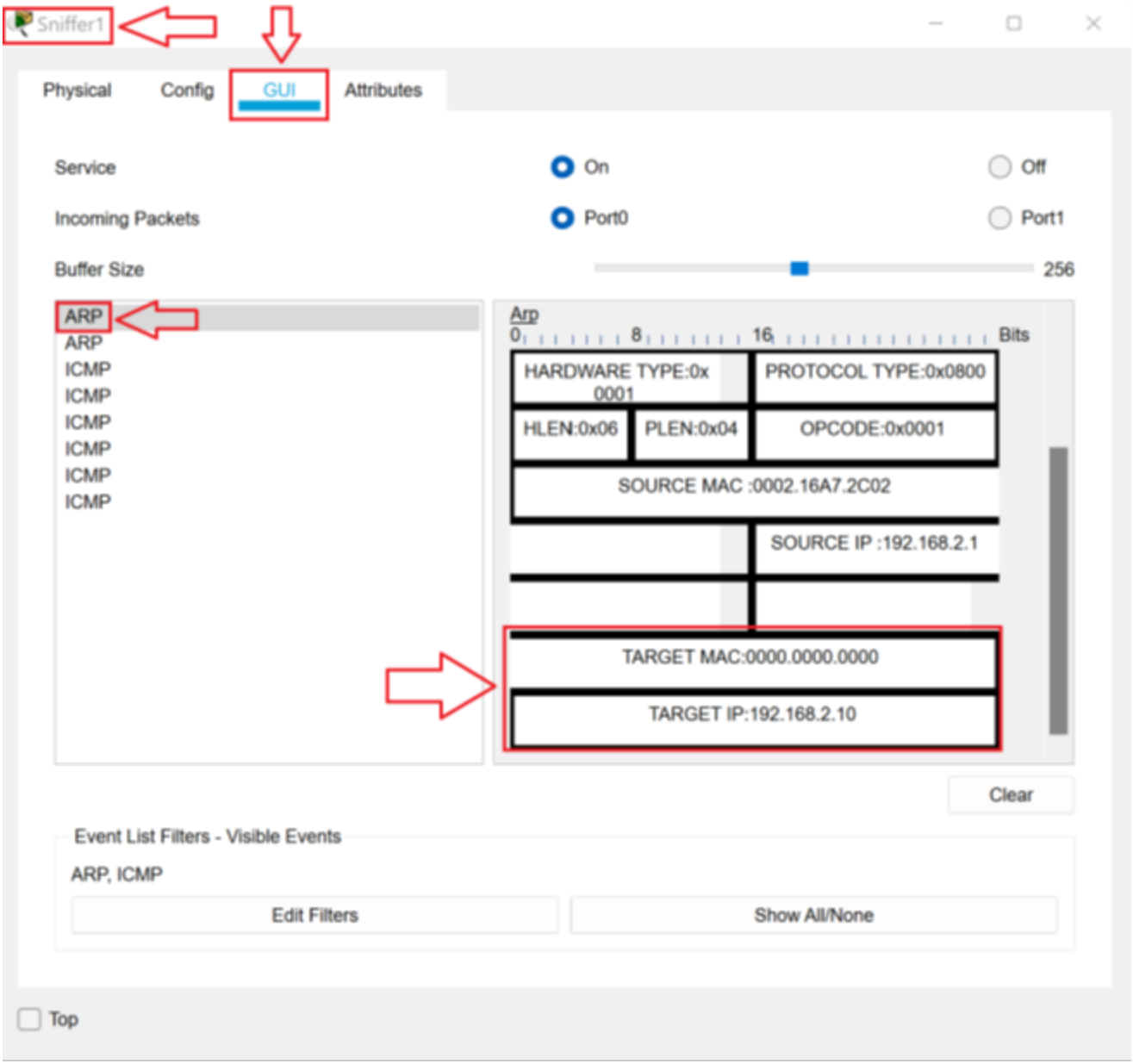
10. Close the **PC0 Properties** dialog box.
11. Click Sniffer0. On the GUI tab, notice the sniffer has captured two ARP packets and eight ICMP packets.
12. Click **Sniffer0**. On the **GUI** tab, you should see two ARP packets and eight ICMP packets. (Don't worry if there are any other packets showing 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**.



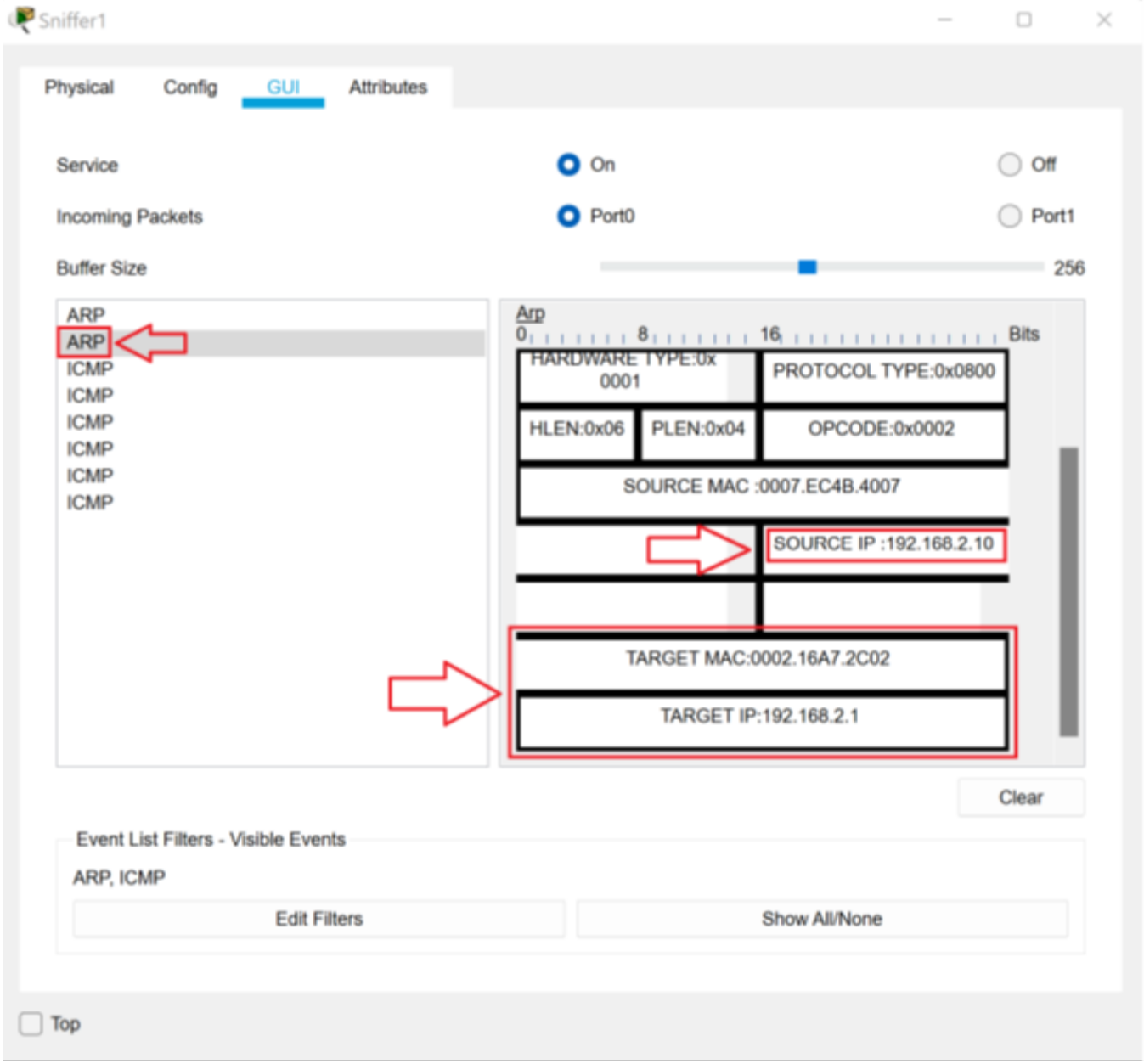
13. Notice that the **TARGET MAC** address is all zeros. The **TARGET IP** address is 192.168.1.1. Since PC0 needs to send the packets to the 192.168.2.0 network, this is the broadcast to find the MAC address of the router.
14. Click on the second **ARP** packet. Scroll down in the ARP packet until you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



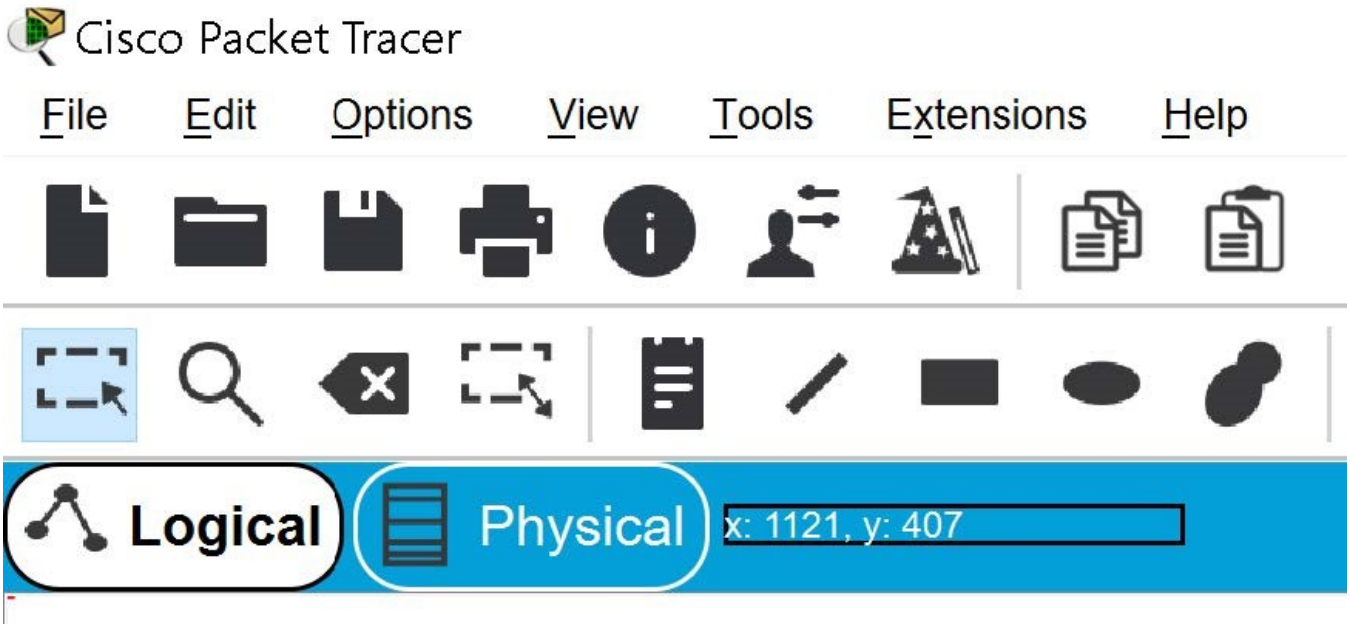
19. Notice that the **TARGET MAC** is the MAC address of 192.168.2.1, which is the **TARGET IP** address. The reply from 192.168.2.10 goes directly to 192.168.2.1. The router knows the MAC address of PC2 by looking at the **SOURCE MAC**. Close the **Sniffer1** Properties dialog box.
20. Close the **3.4.3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

Configure Wireless

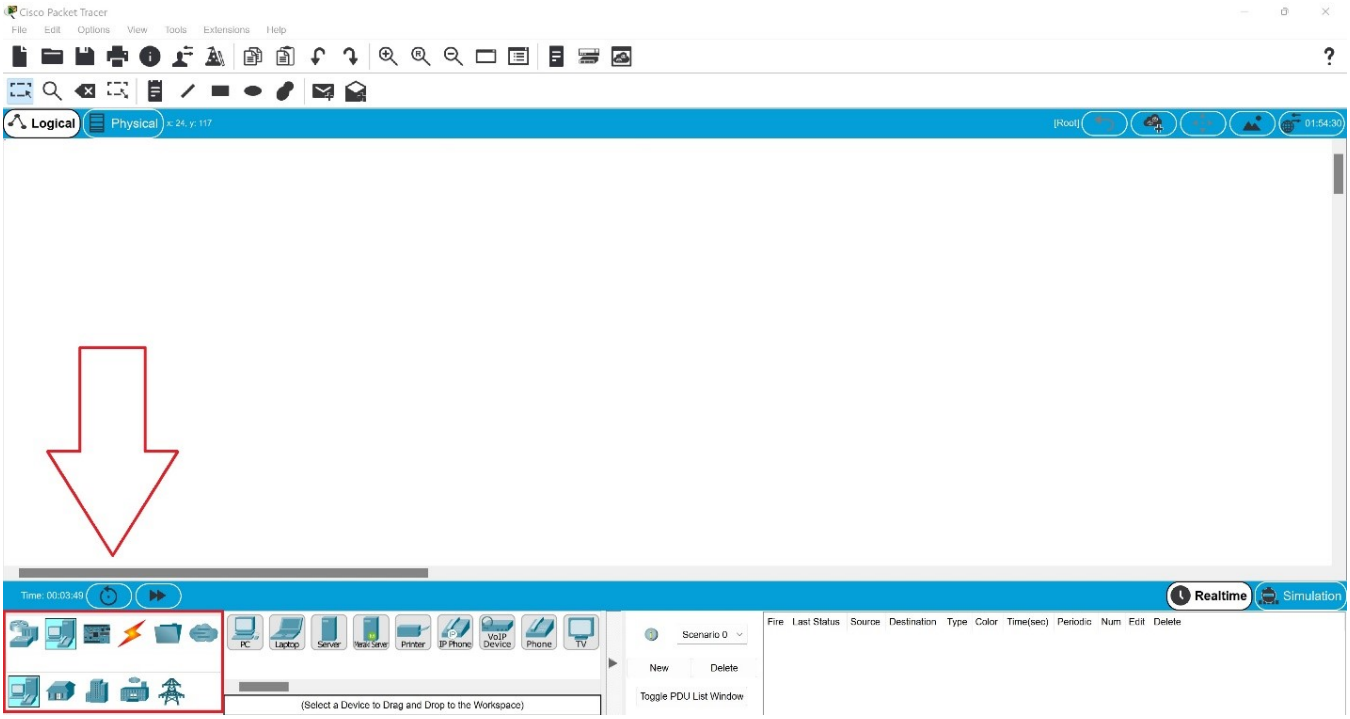
In this lab, we will set up a small wireless network.

TASK A

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

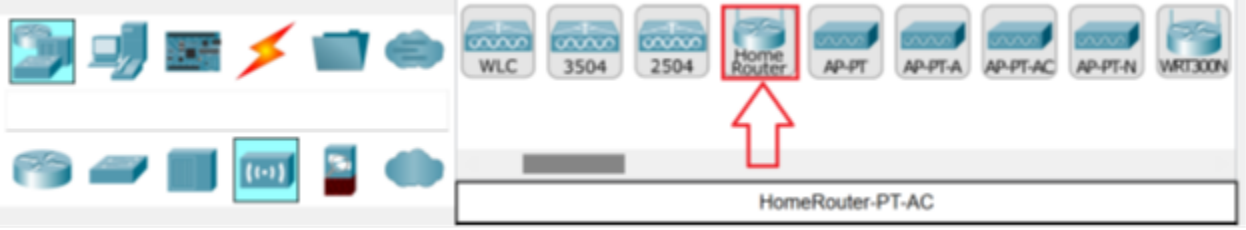


3. In the bottom left corner of the screen, locate the **toolbox area**.

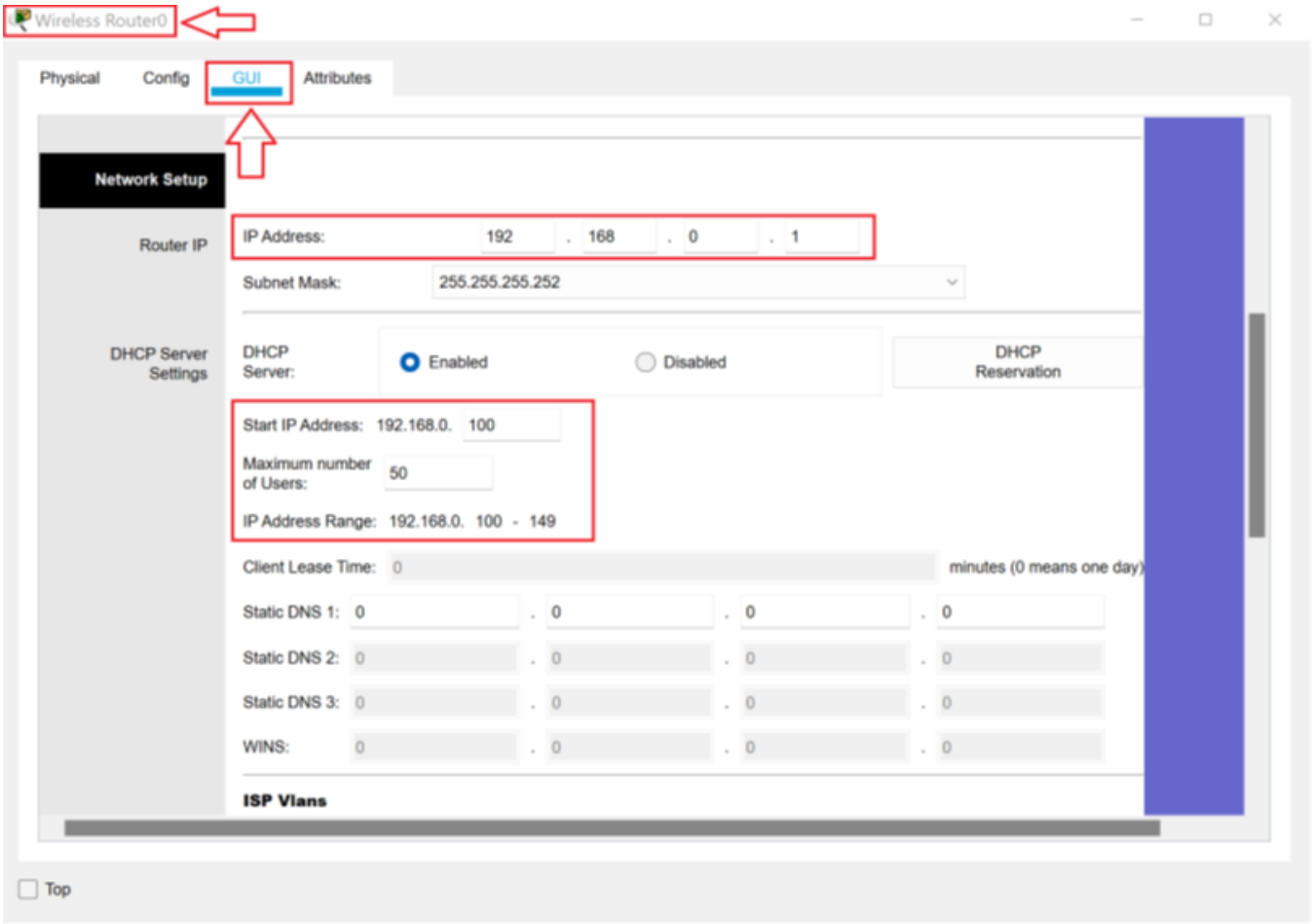


4. In the **toolbox area**, select **Network Devices** and then select **Wireless Devices**.

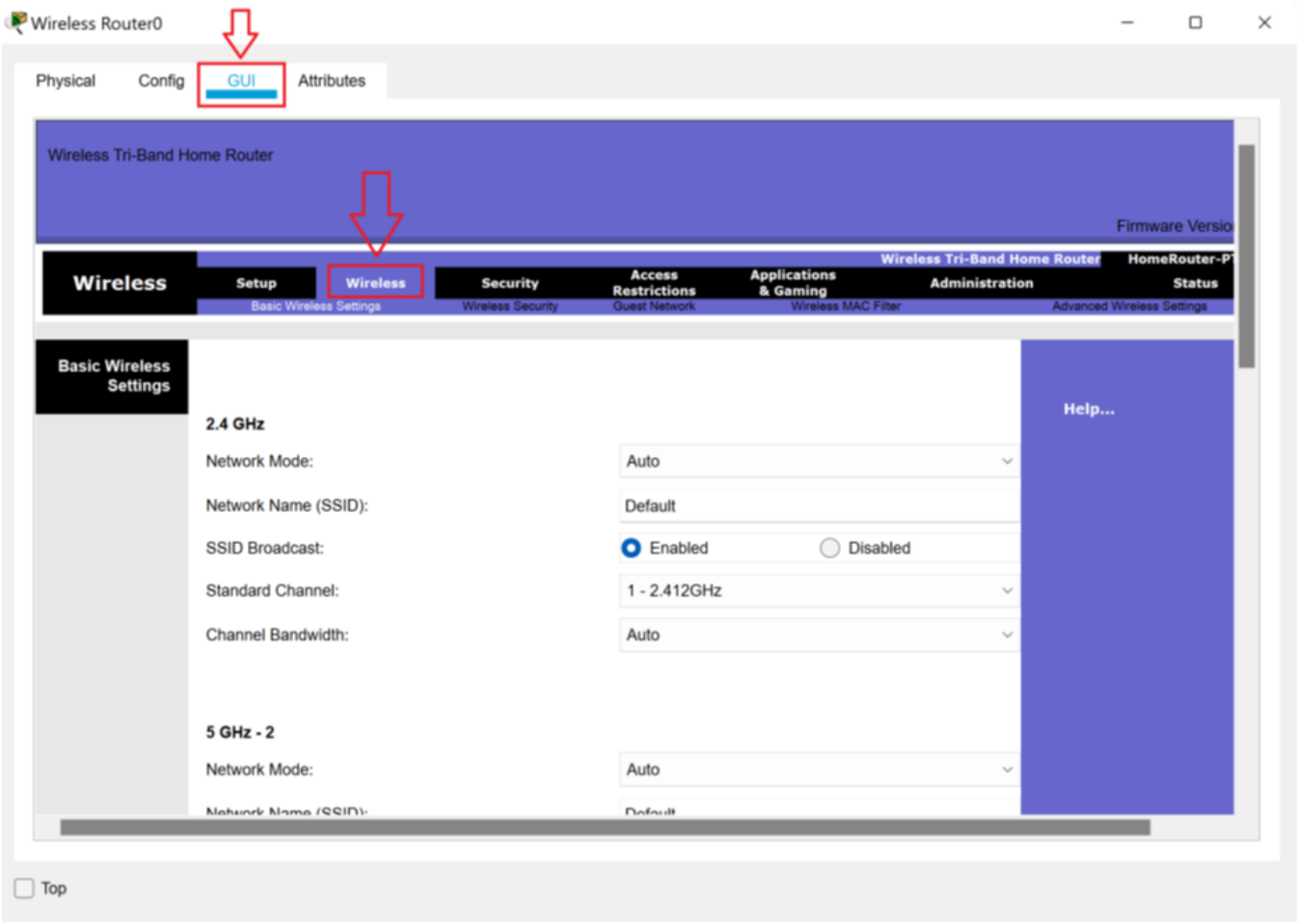
5. Drag a **Home Router** onto the logical diagram.



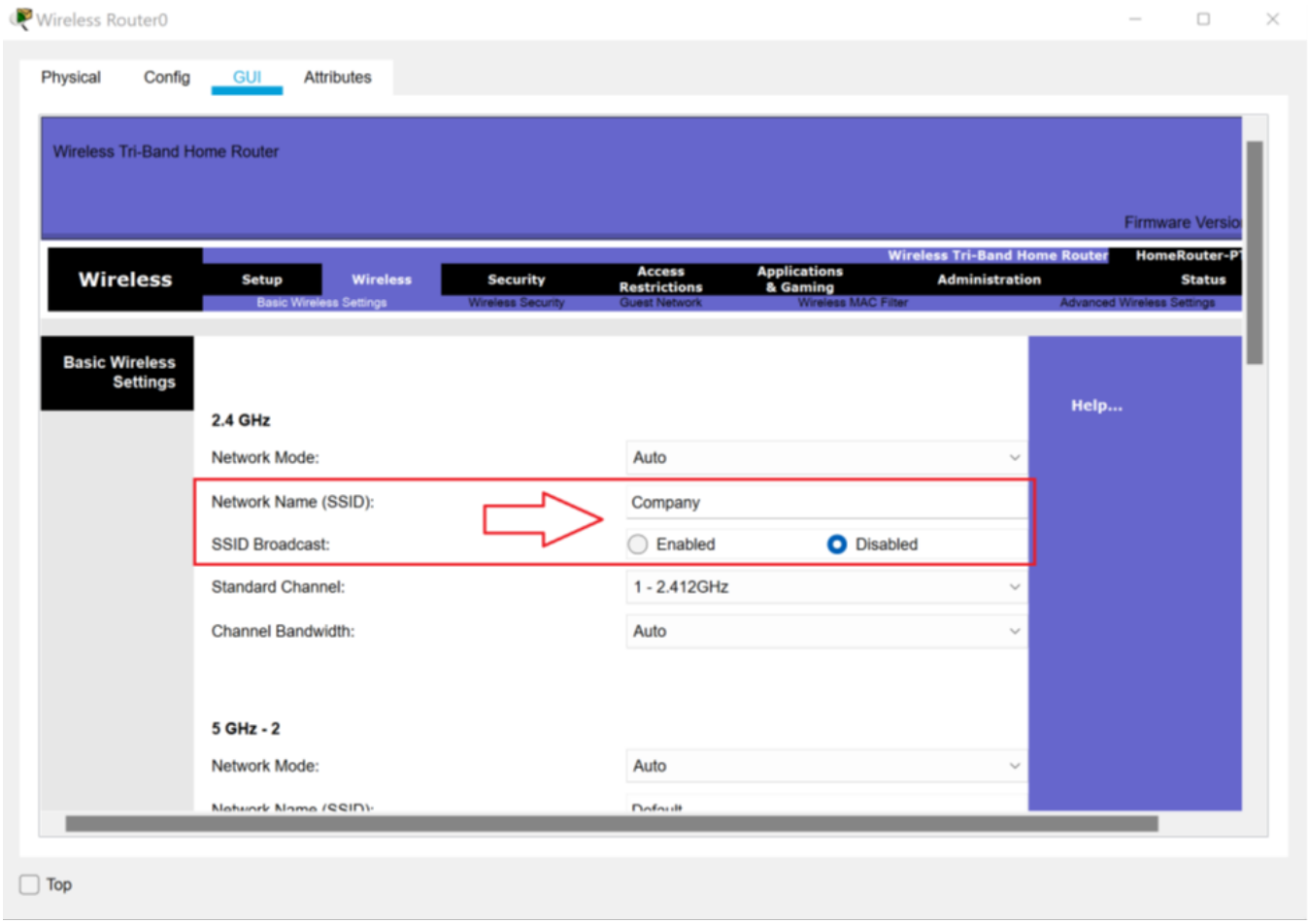
6. Click **Wireless Router0** to open the **Wireless Router0 Properties** dialog box. Then click the **GUI** tab. Scroll down to the **Network Setup** section.



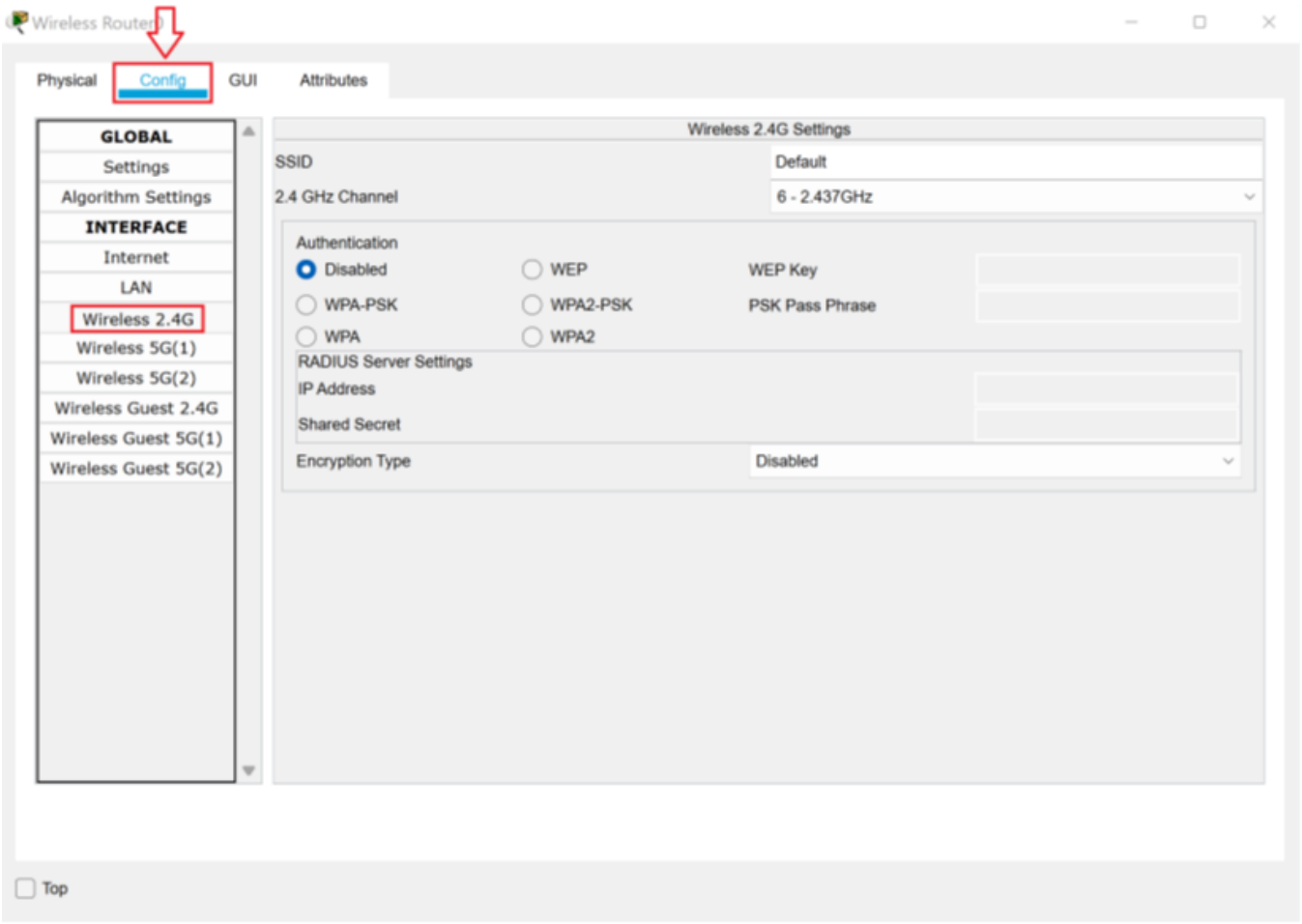
7. Notice the router's IP address is 192.168.0.1. The router will function as a DHCP server. It can give out 50 IP addresses starting with 192.168.0.100.
8. In the **GUI** tab, click the **Wireless** menu option.



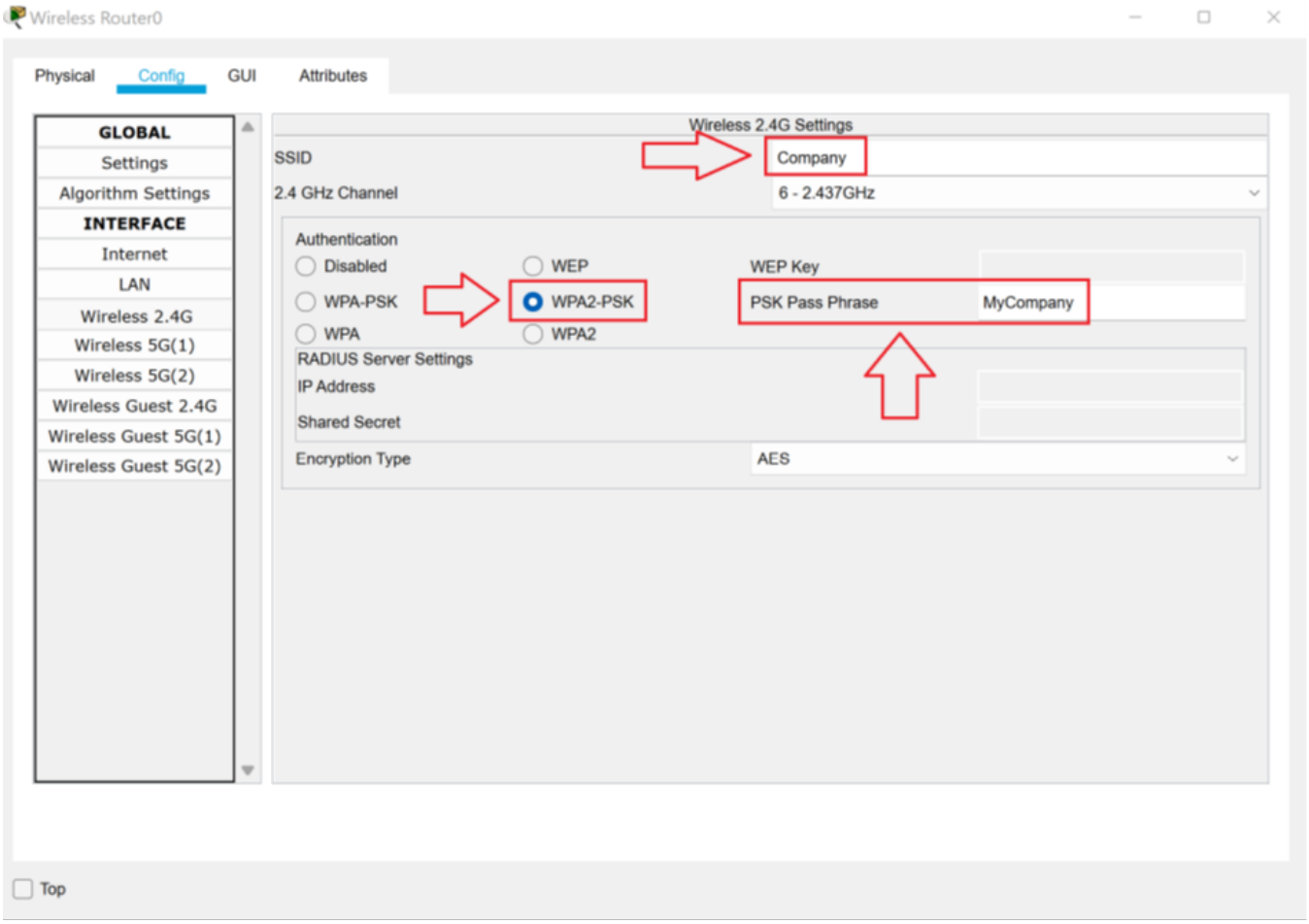
9. The wireless clients will connect to the wireless routing using the 2.4 GHz band. Change the **Network Name (SSID)** to **Company** and set **SSID Broadcast** to **Disabled**.



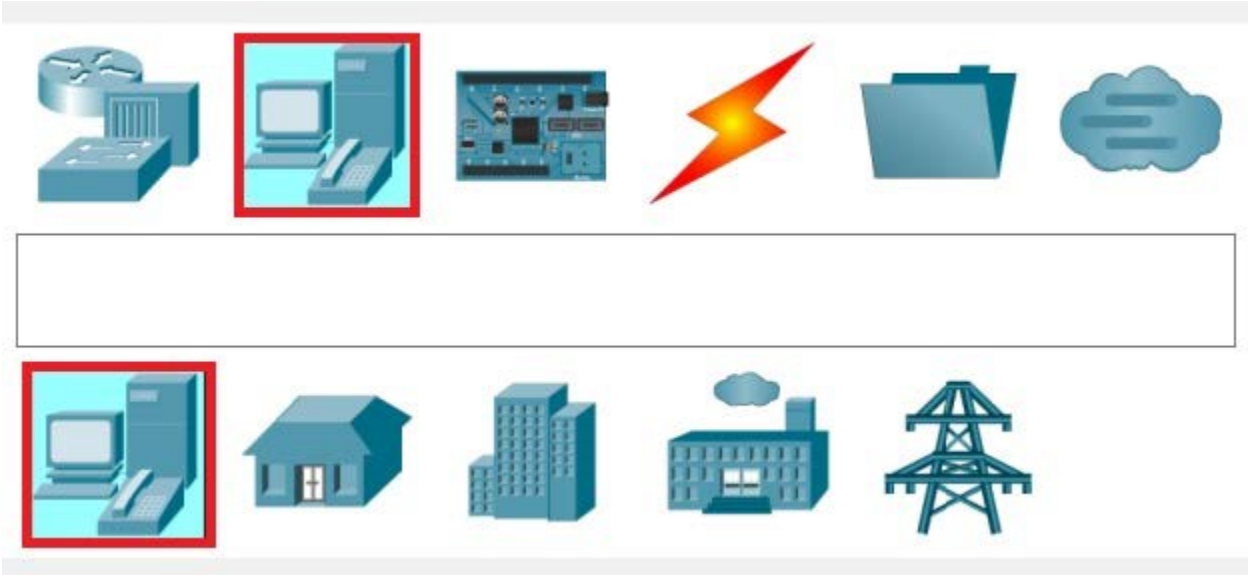
10. In the **Wireless Router0 Properties** dialog box, click the **Config** tab. In the **Interface** menu, click **Wireless 2.4G**.



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**.

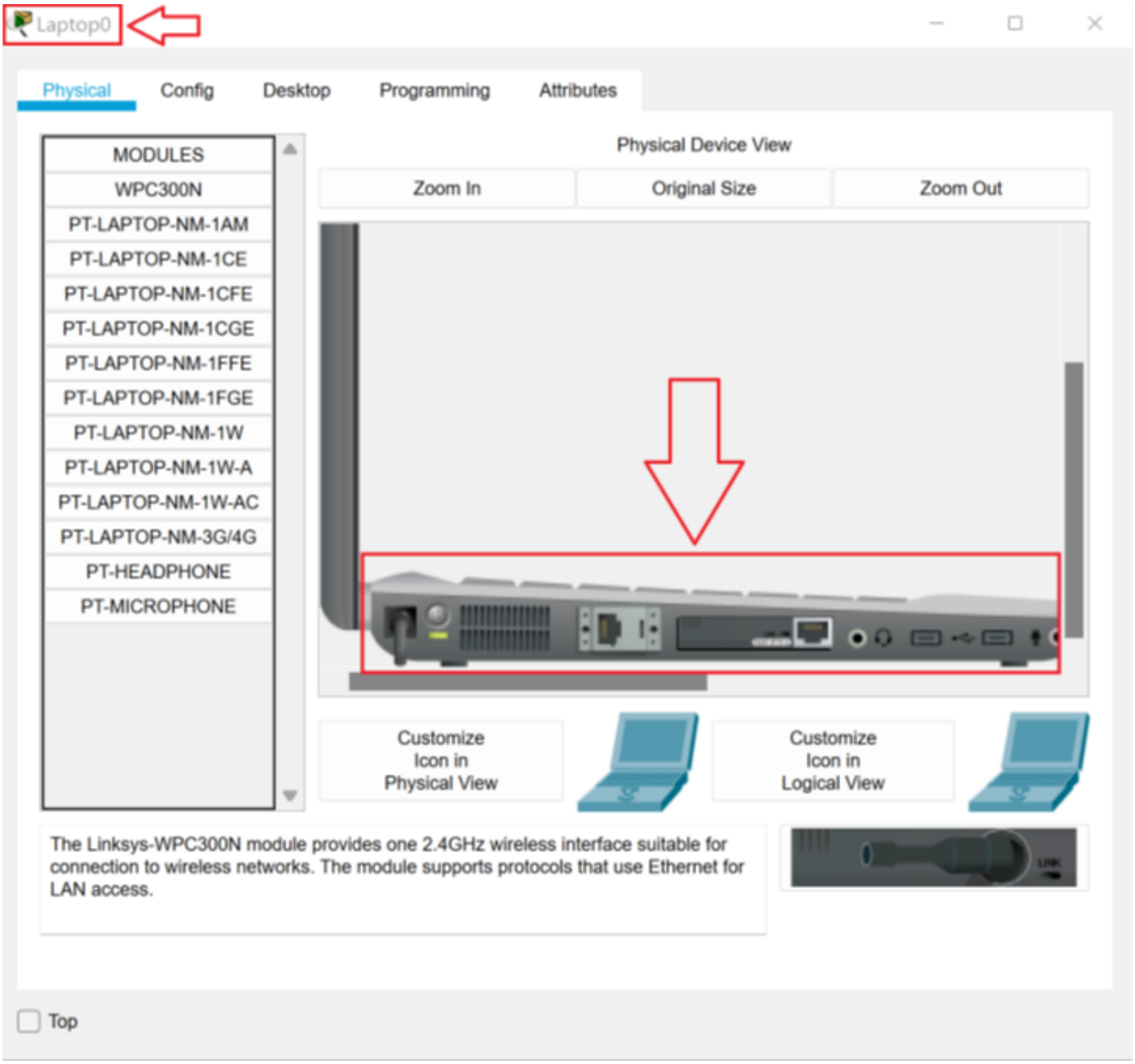


14. Drag a **Laptop** onto the logical diagram.

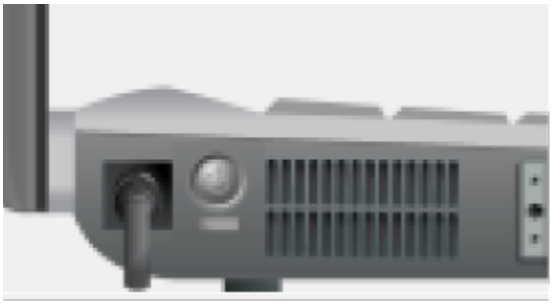
HomeRouter-PT-AC
Wireless Router0

Laptop-PT
Laptop0

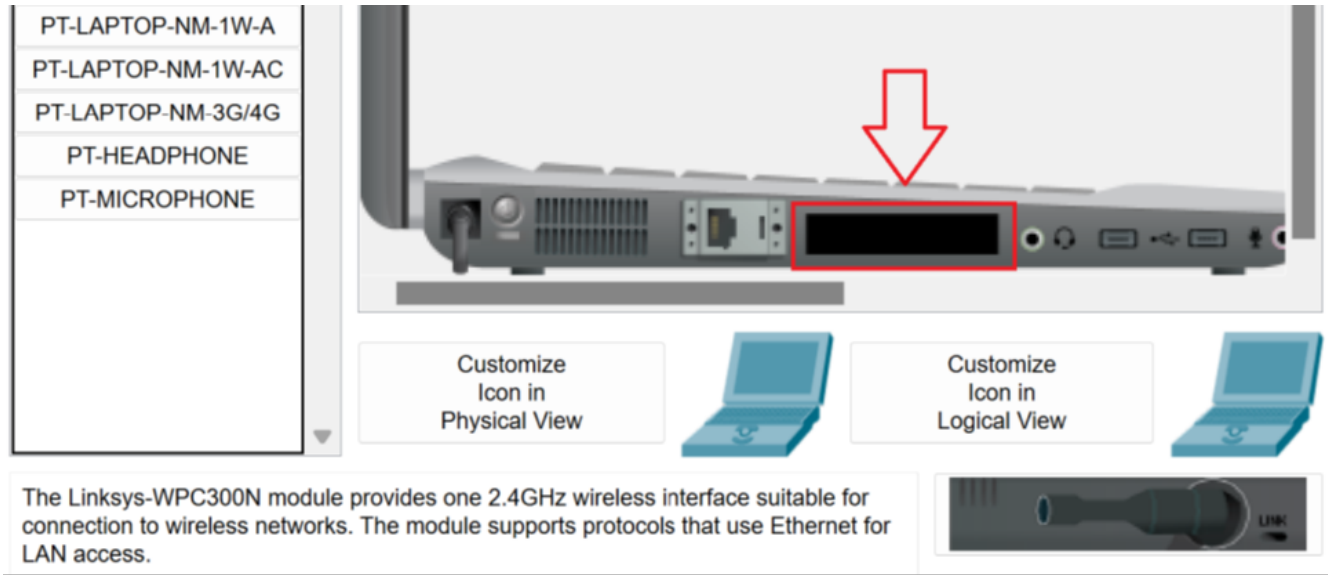
15. Click on **Laptop0** to open the **Laptop0 Properties** dialog box. Scroll down so you can see the side of the laptop.



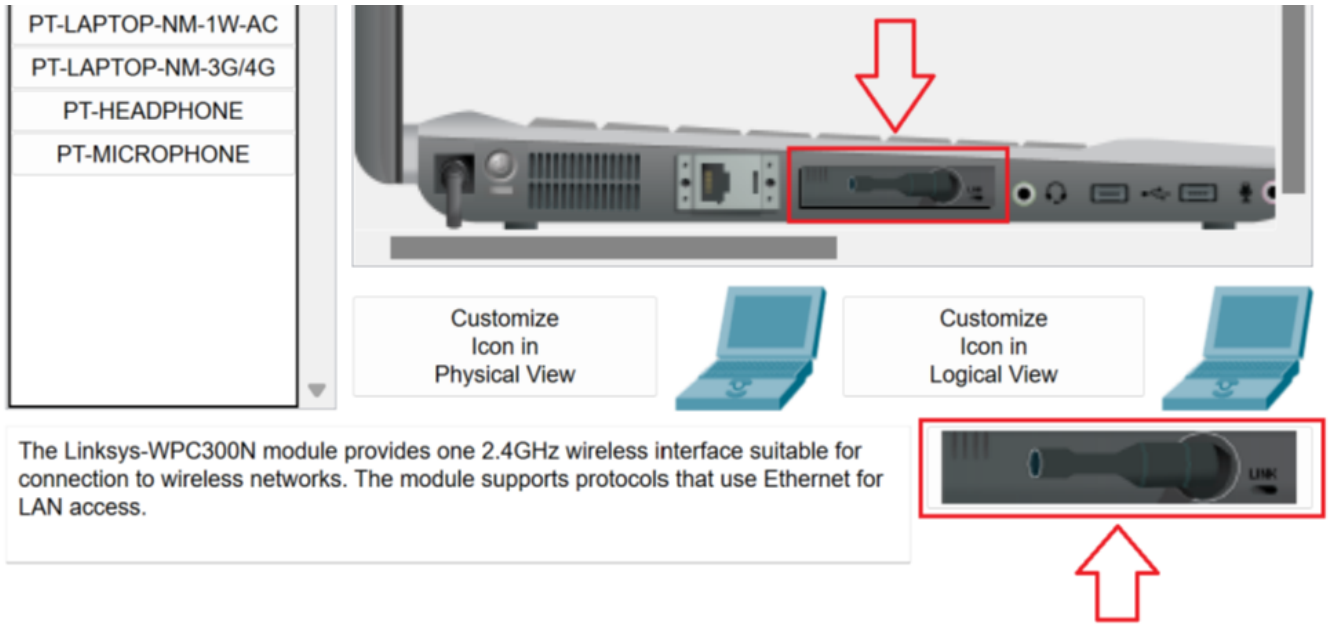
16. The laptop comes configured with a wired network card. To use the wireless, we need to change out the network card. That can only be done when the laptop is off. To power off the laptop, click the power button. The power light will change from green to gray.



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

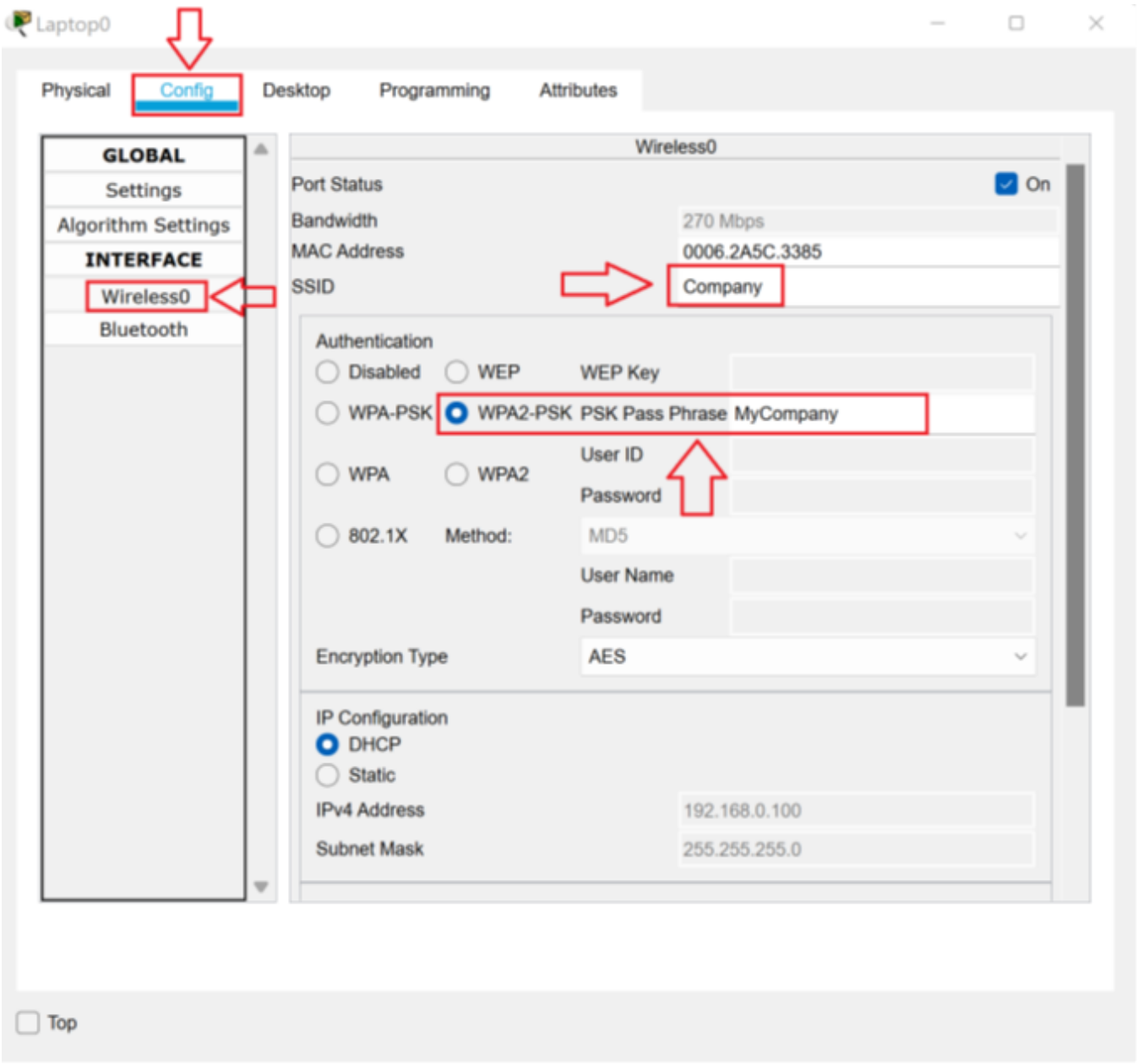


18. From the bottom right of the **Laptop0 Properties** dialog box, drag the wireless adapter to the empty network card slot.

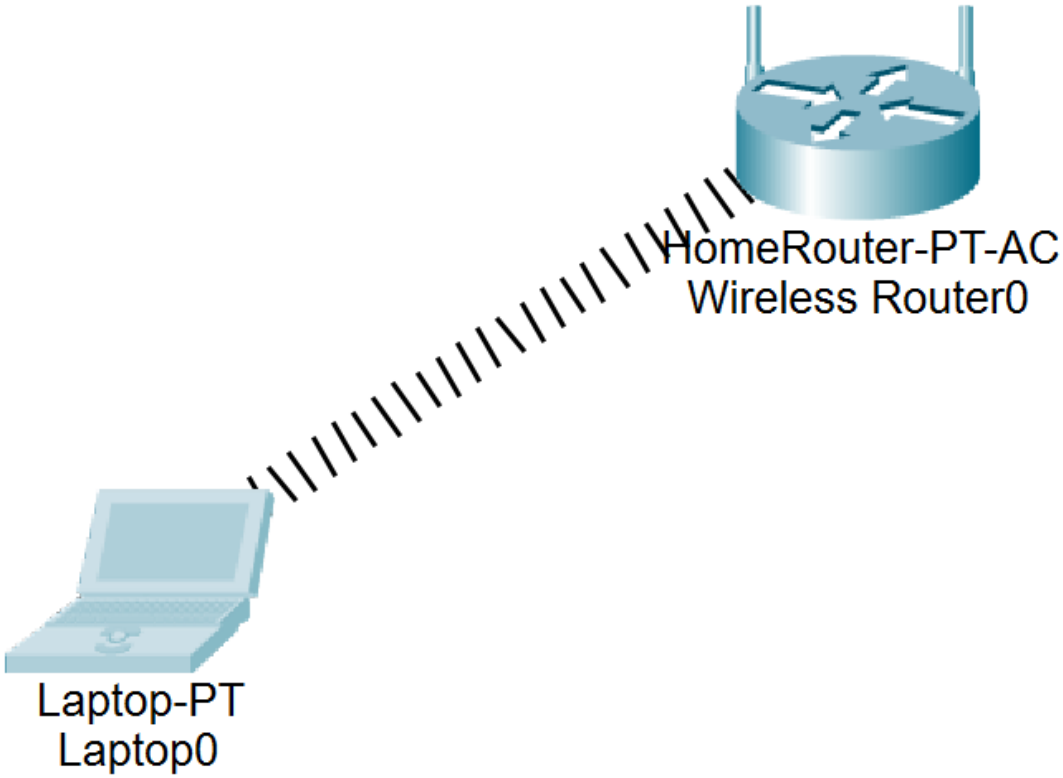


19. Click the power button to power on the laptop. The power light will change from gray to green.

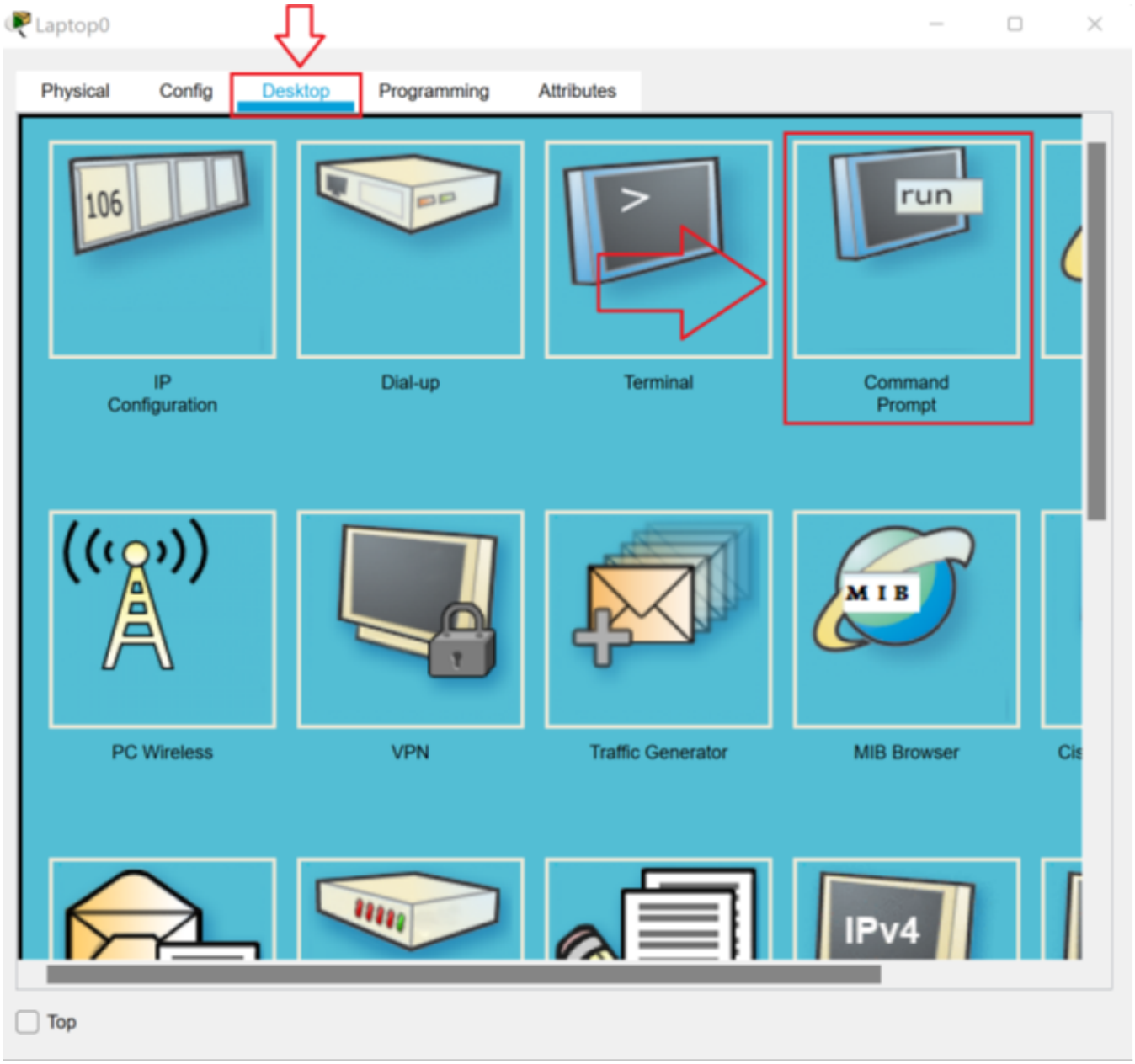
20. In the **Laptop0 Properties** dialog box, click the **Config** tab. In the **Interface** menu, click **Wireless0**. Change the SSID to **Company**. Select the **WPA2-PSK** radio button. In the **PSK Pass Phrase** text box, type **MyCompany**.



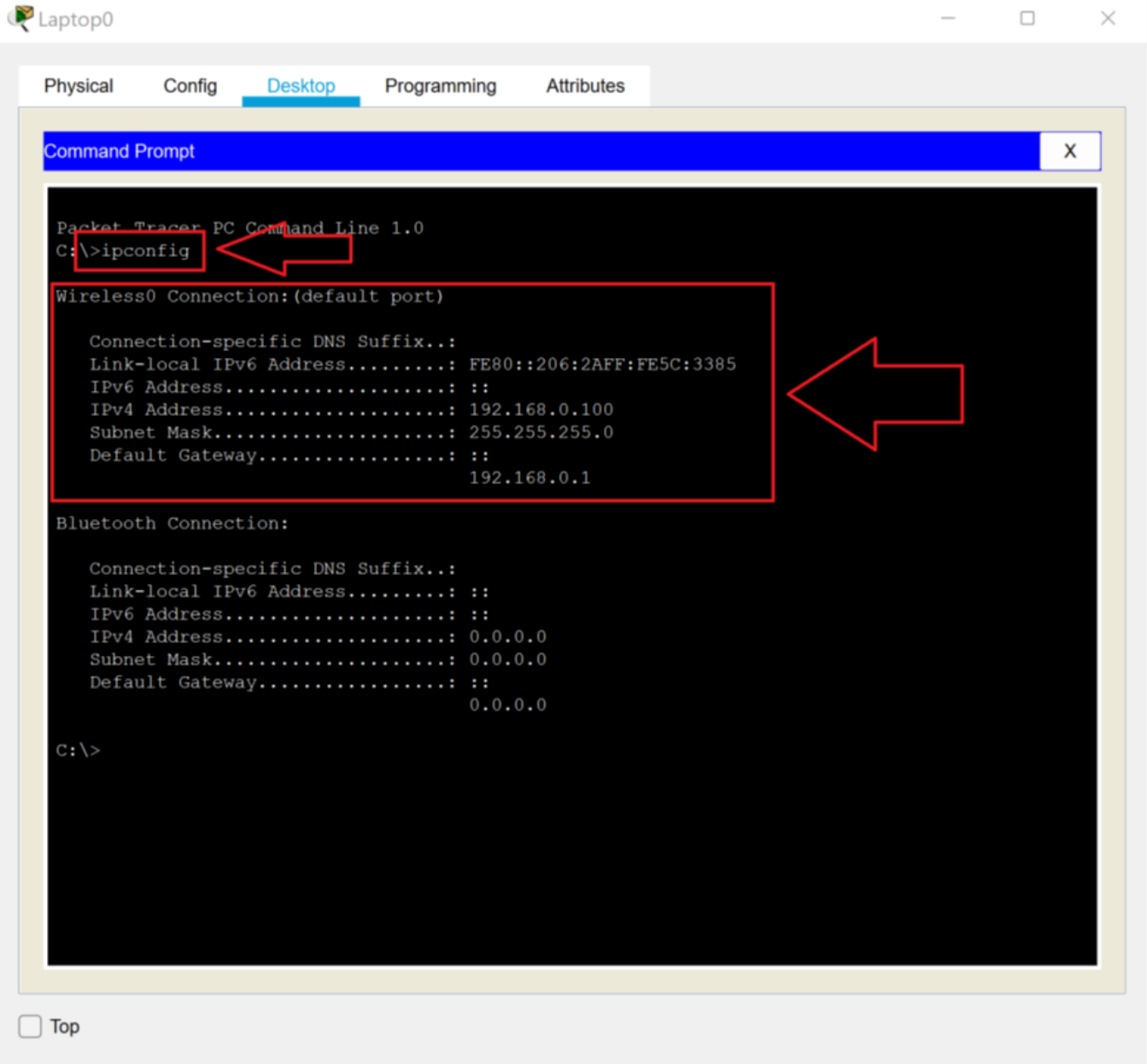
21. In the logical diagram, the laptop will show a wireless connection with the router.



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.

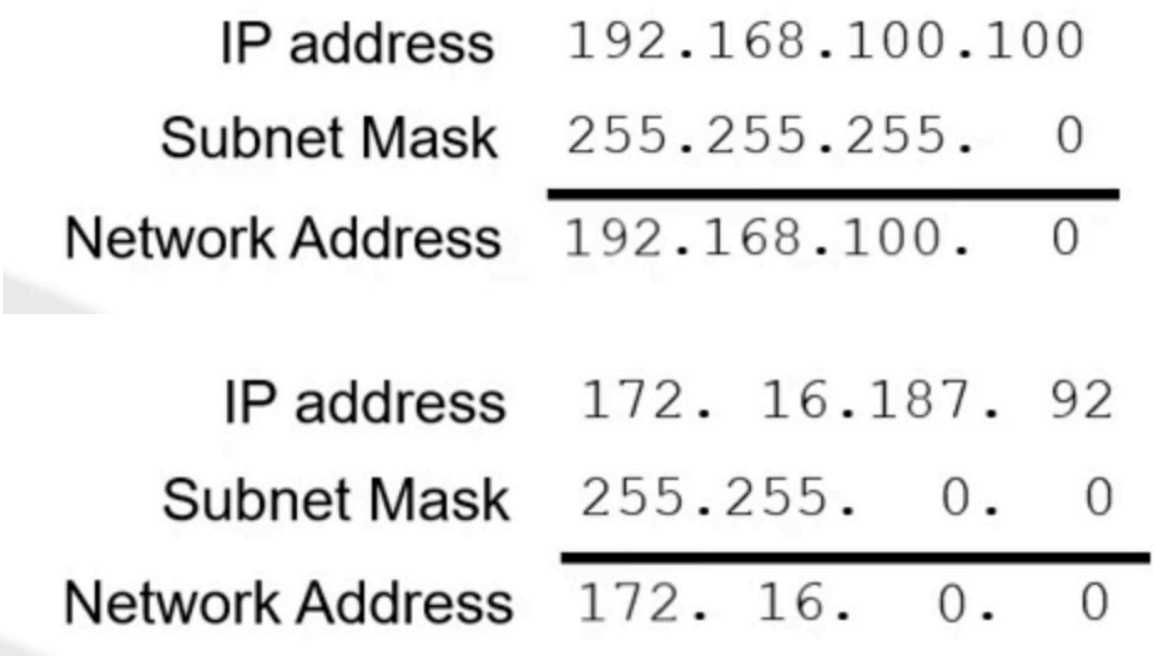
If we apply this analogy directly to TCP/IP, the local network would be like a “country.” Each network has a network address that works like a country code. Each node on the network has a node address that identifies the node. Millions, maybe billions, of nodes have the same node address. But on each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

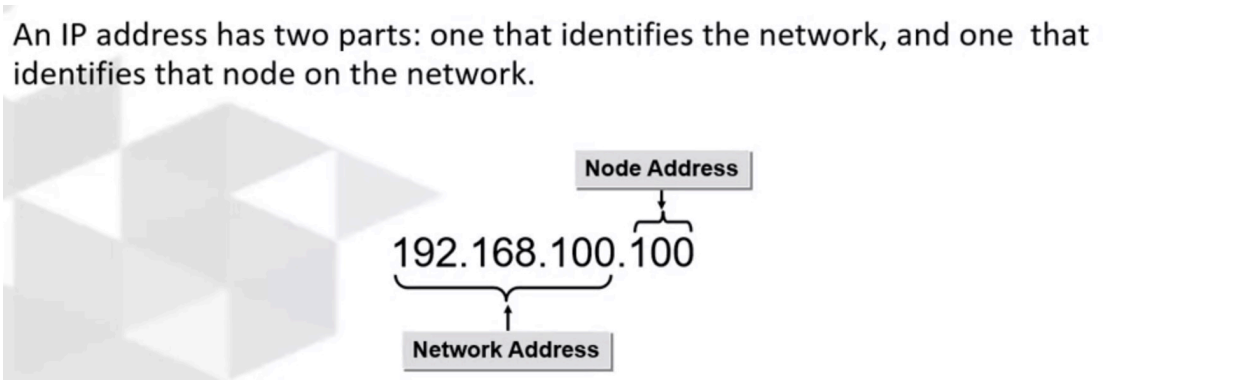
Any number in the IP address that's in the same position as a 255 in the subnet mask is part of the network address. Any number that's in the same position as a zero is part of the host address. The process computers used to compare the IP address to the subnet mask to find the network address is called

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.

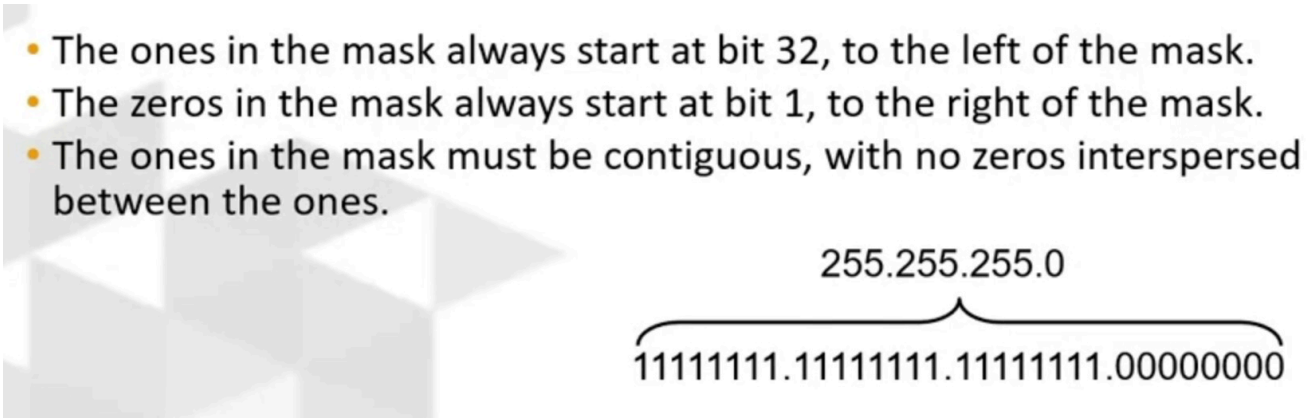


Subnet Mask

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.

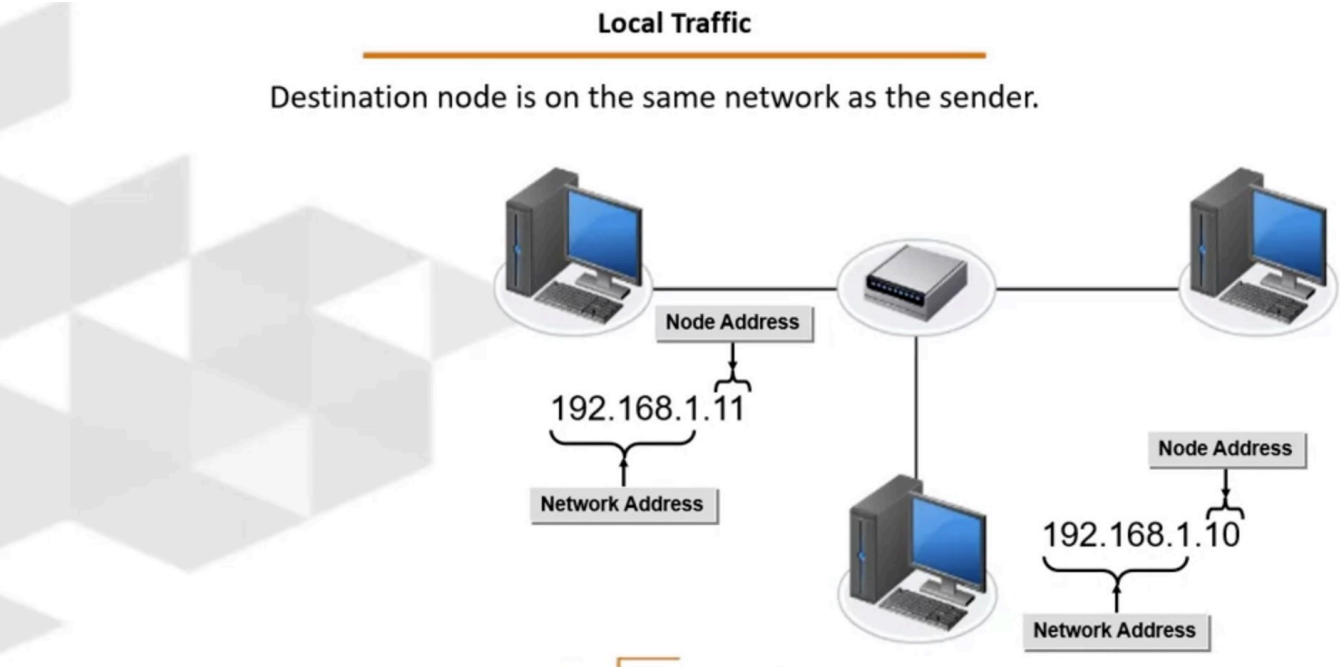


The subnet mask is a 32-bit number, also written in dotted decimal form, four octet will only see the numbers 0 through 255. The difference between the IP address and the subnet mask is that in the subnet masks, all the ones in binary are on the left side. All the zeroes in binary are on the right side, 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.



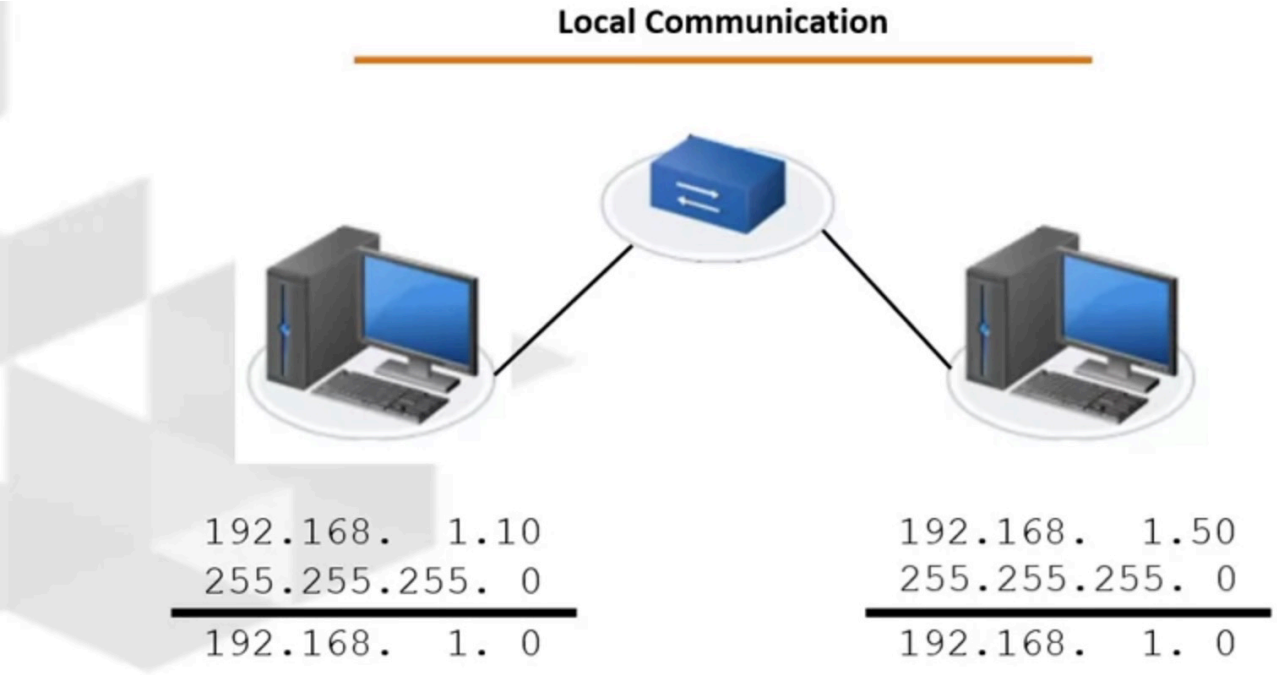
Local or Remote? Part1

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?



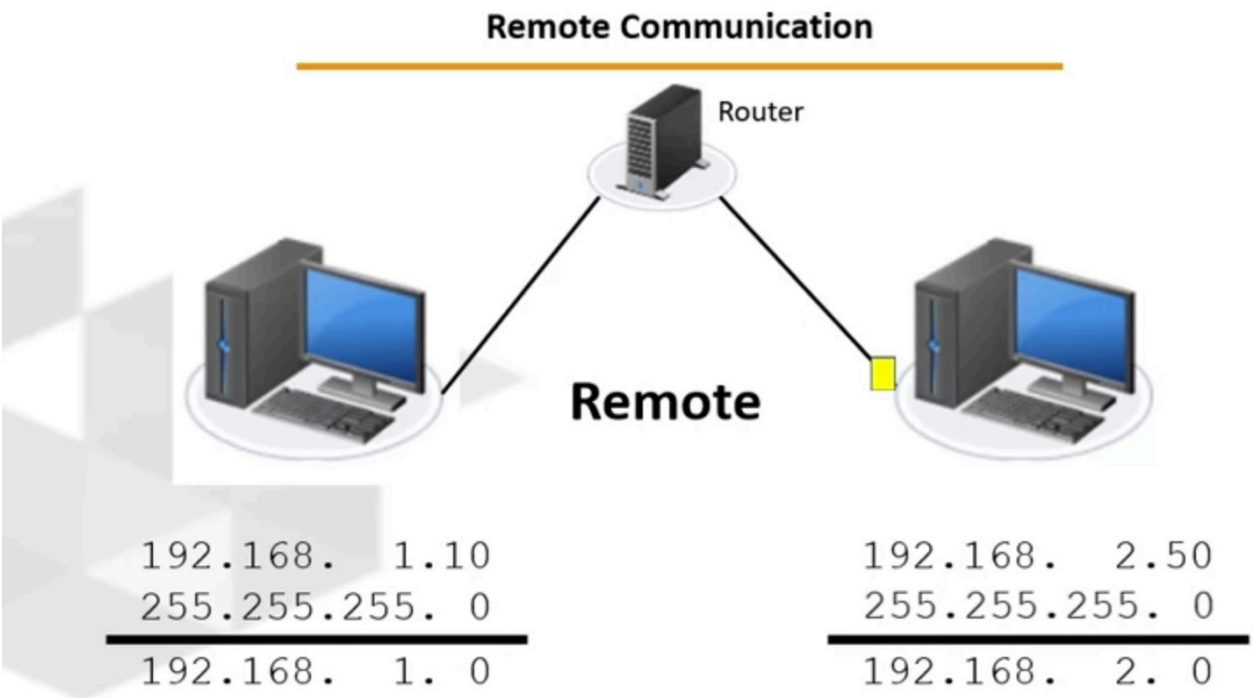
Let's take a look at an example. Suppose I'm sitting at a sender on the left and it has an IP address of 192.168.1.10 with a sub net mask of 255.255.255.0. And this computer is going to communicate with another computer that has the IP address of 192.168.1.50. The sending device first needs to figure 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.

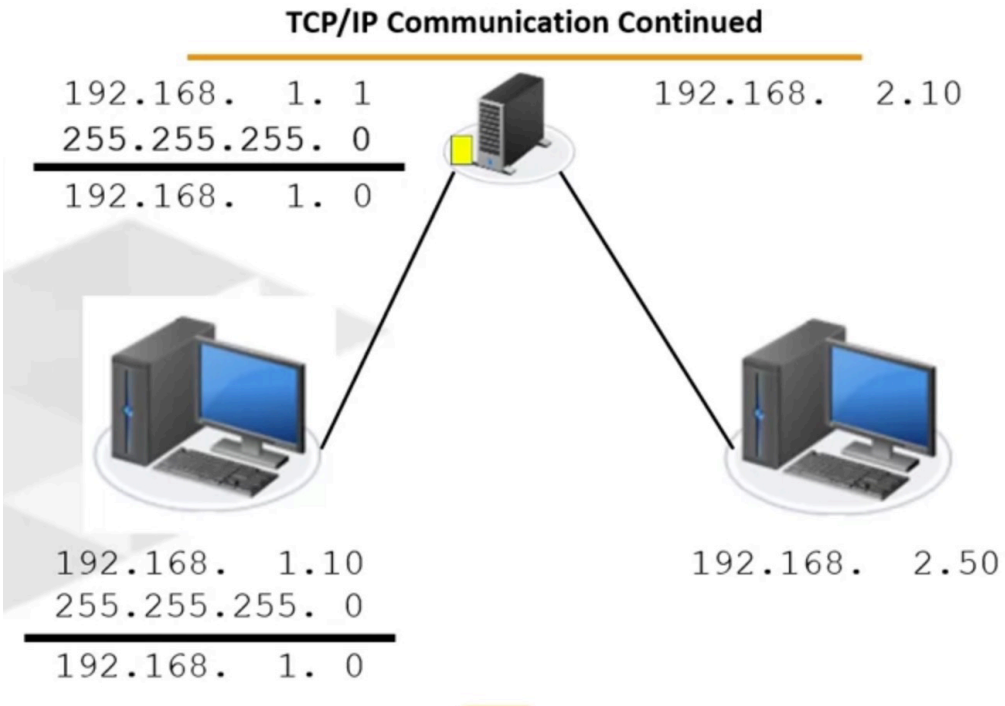


Local or Remote? Part2

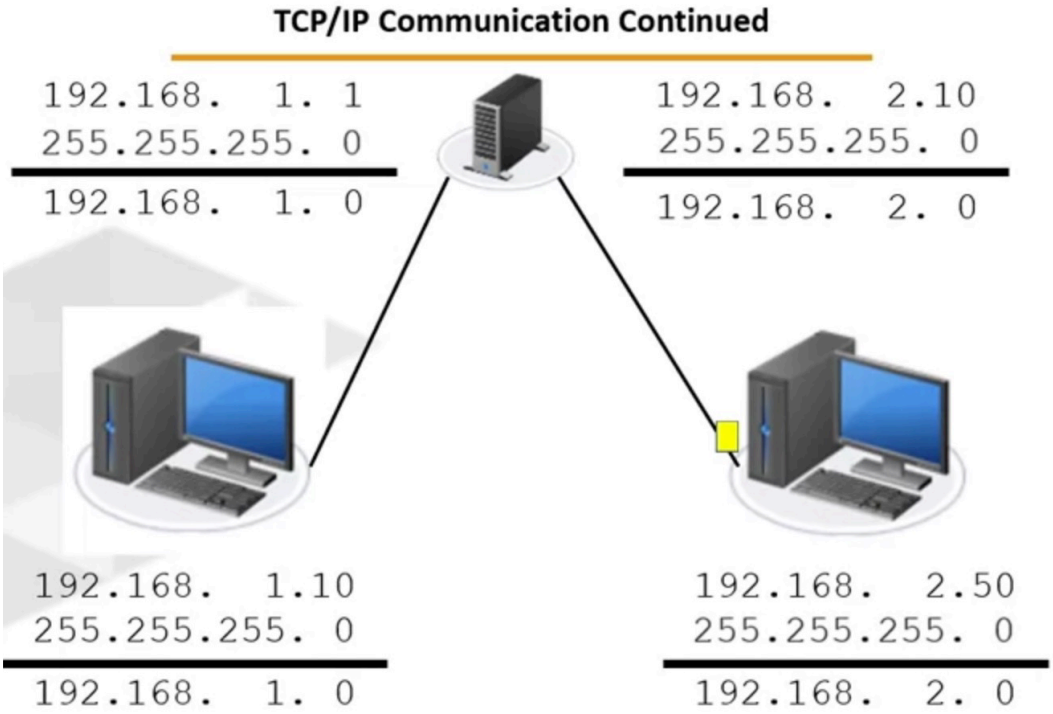
If the sender and the receiver have different network addresses, they are on different networks. Datacenter device on a different network is called remote. Let's take a look at an example. Suppose the sender has an IP address of 192.168.1.10, and a subnet mask of 255.255.255.0, and it wants to send data 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.

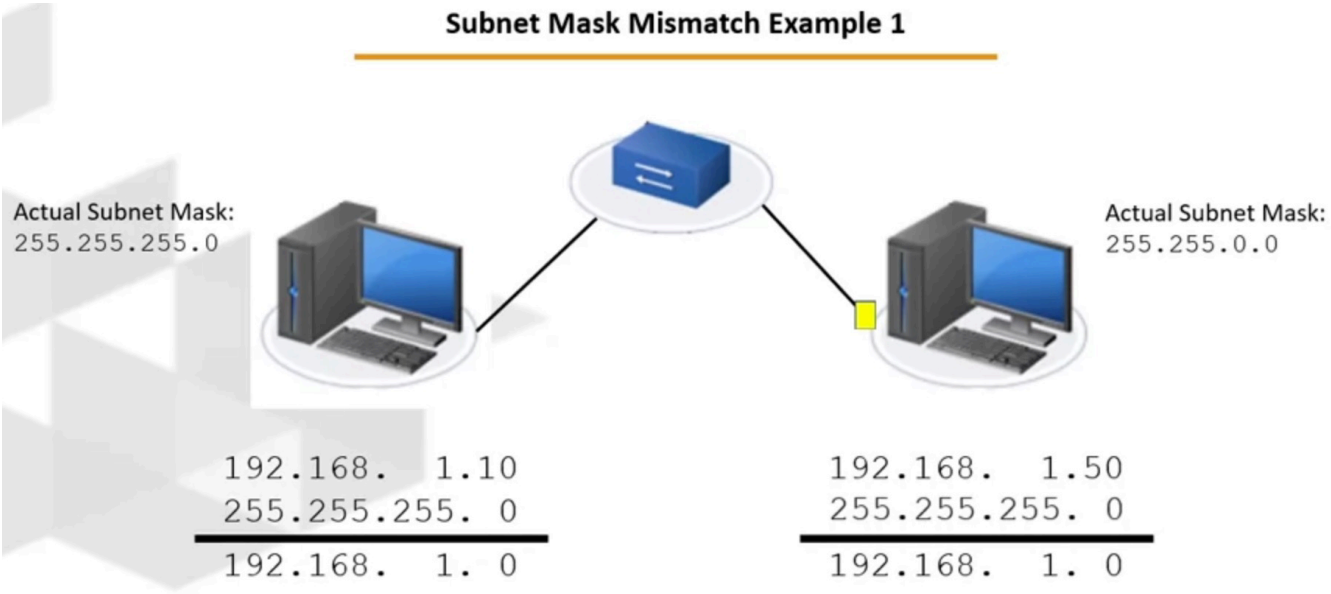


Now, watch out. When two IP addresses are remote. Always check the network address of the router. I can't tell you how many people I see that don't do that. Don't assume the router is correctly configured either on the client or on the router side. If something isn't working, it means there's a problem. 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.

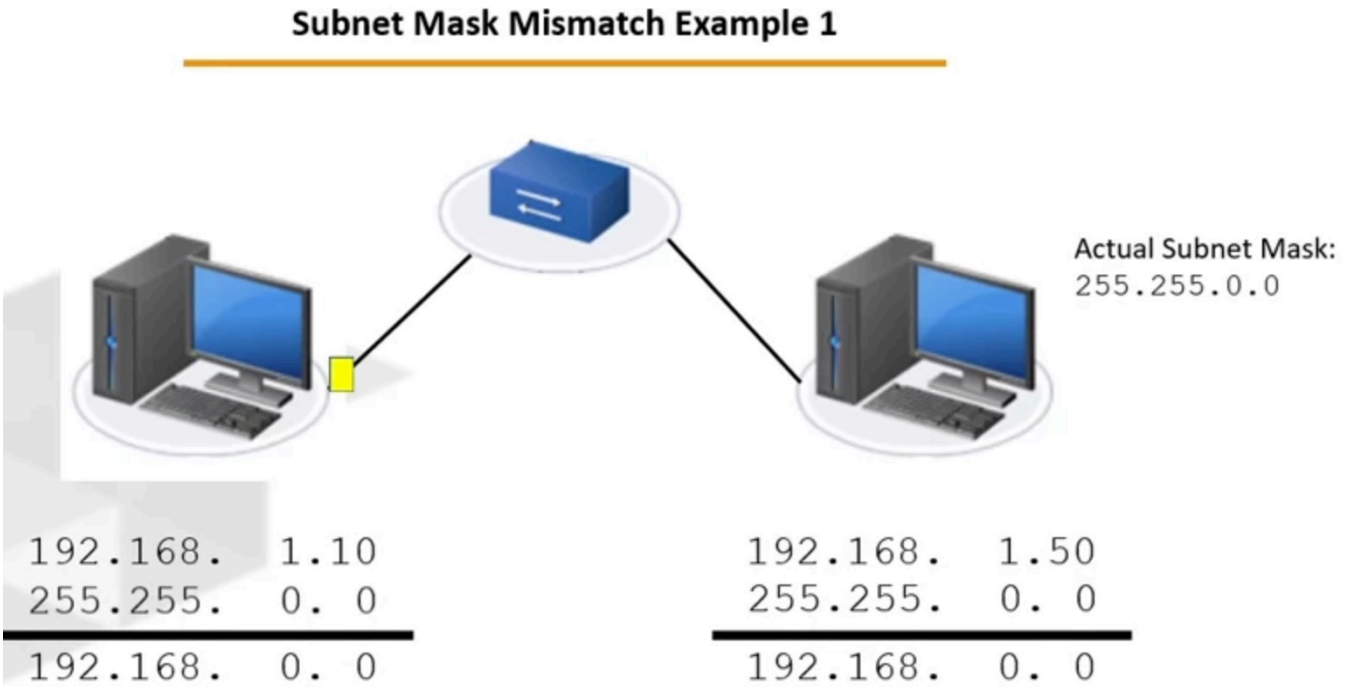
Local or Remote? Part3

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.

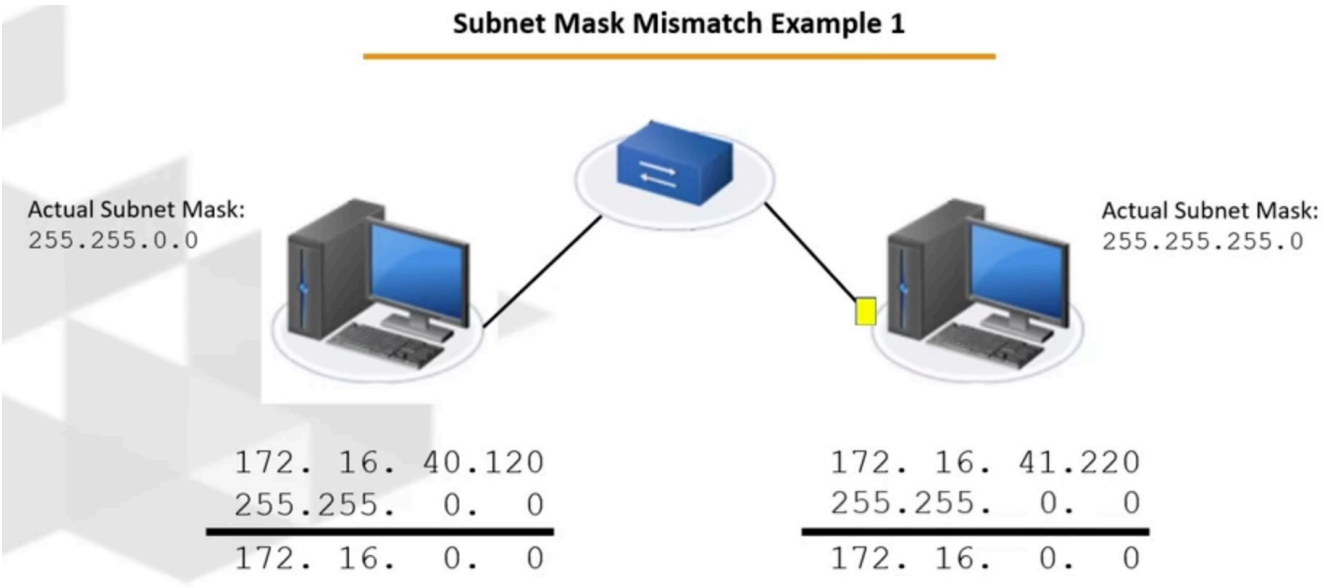
The answer to that depends on the IP addresses and the subnet masks being used. So we're going to look at a couple of examples. So for the first example, we're going to take our 192.168.1.10 computer with a subnet mask of 255.255.255.0. And it wants to communicate with 192.168.1.50 but that actually 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.



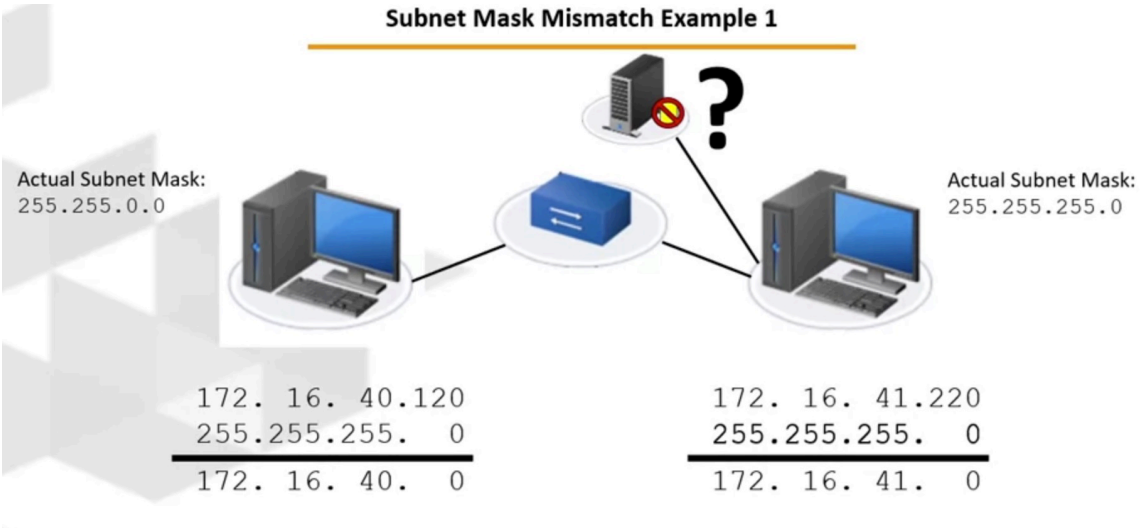
And if it's going to reply, then in that case now 192.168.1.50 is our sender and we're going to reverse the process. So 192.168.1.50 takes its subnet mask, the 255.255.0.0. So I'm on network 192.168.0.0, I wonder what network 192.168.1.10 is on? Well, let me use my subnet mass to evaluate, and look they're 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.

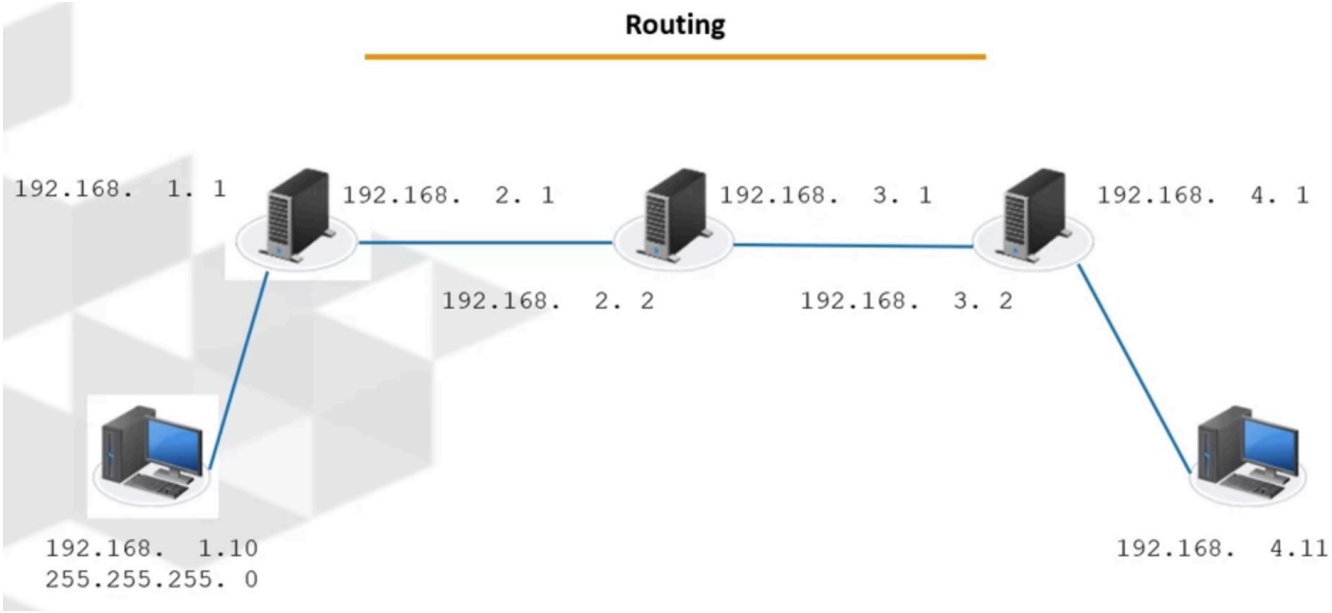


So now the other computer 172.16.41.220 gets the data and now it wants to reply. So we're going to have to do the same thing but we're going to have to do it in reverse, right? So that computer says, all right, my IP address 172.16.41.220, my subnet mask 255.255.255.0, what network am I on? I'm on the 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.

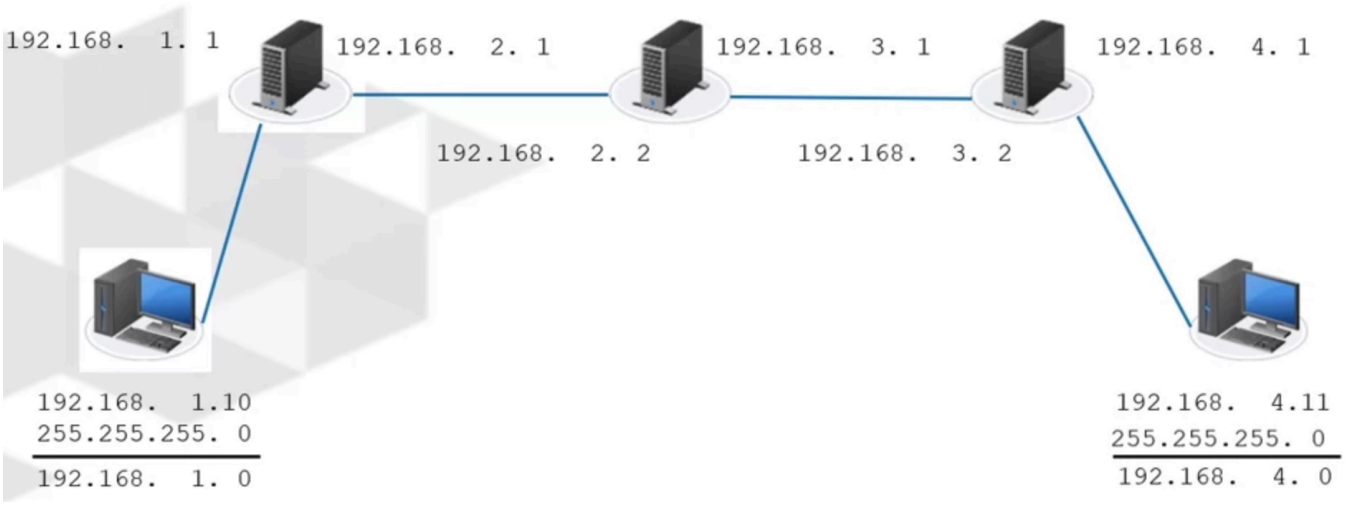


Routing

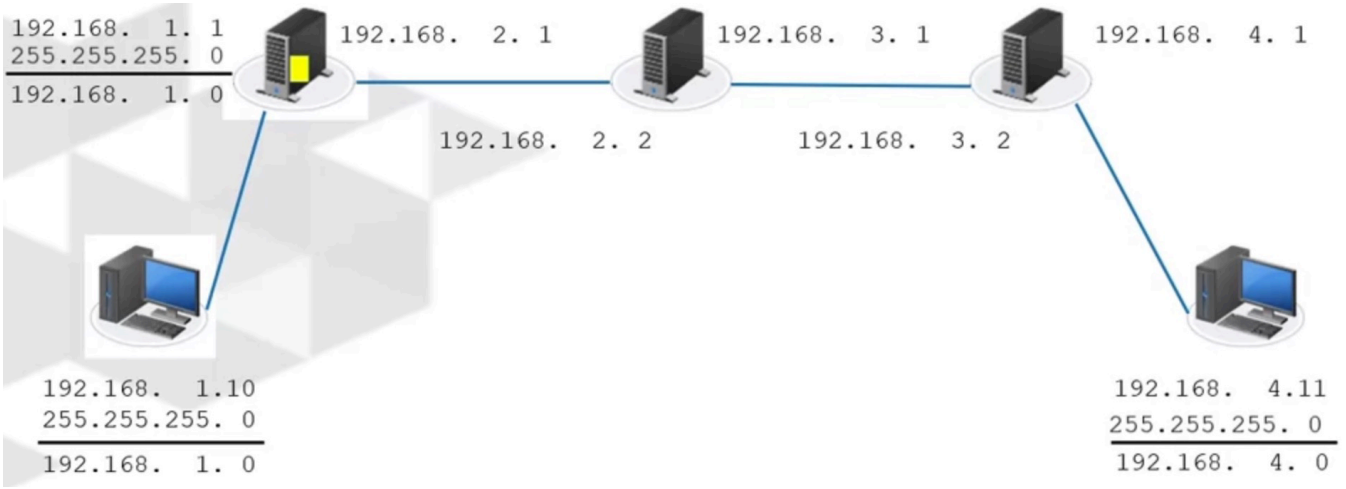
ou know how to find the network ID and you know how to answer that very important question, local or remote. Now it's time to take a look at what actually happens when there's more than one router in between the sending device and the receiving device. I've put together a very simple network for you. 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.



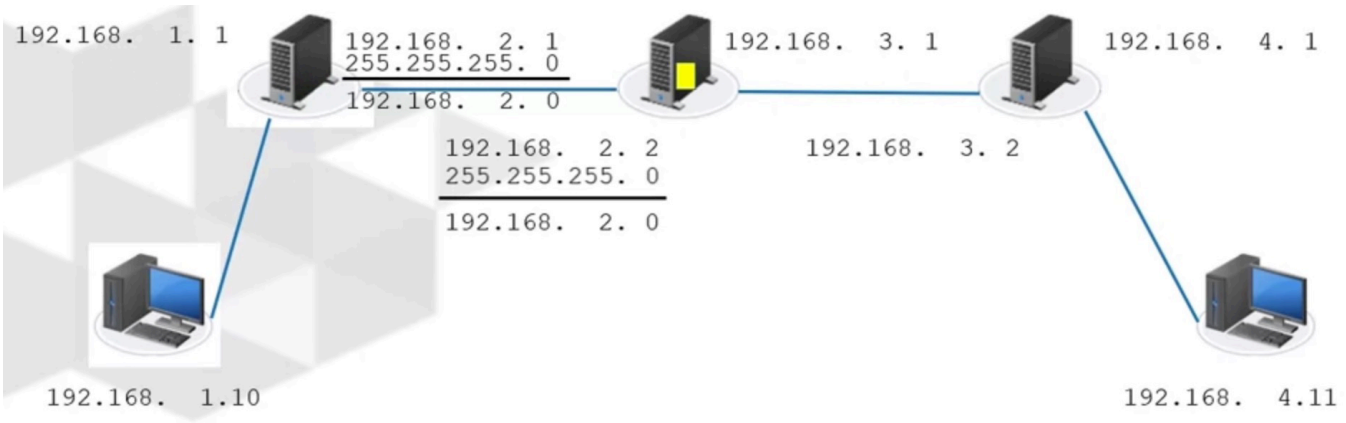
Let's take a look at what actually happens when 1921681.10 wants to send a message to 1921684.11. Of course, we start out with our sender. What network I'm I on? It's going to do its basic ending and say, hey, I'm on network 1921681.0. Gee, I wonder what network 1921684.11 is on. Remember it uses its 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.



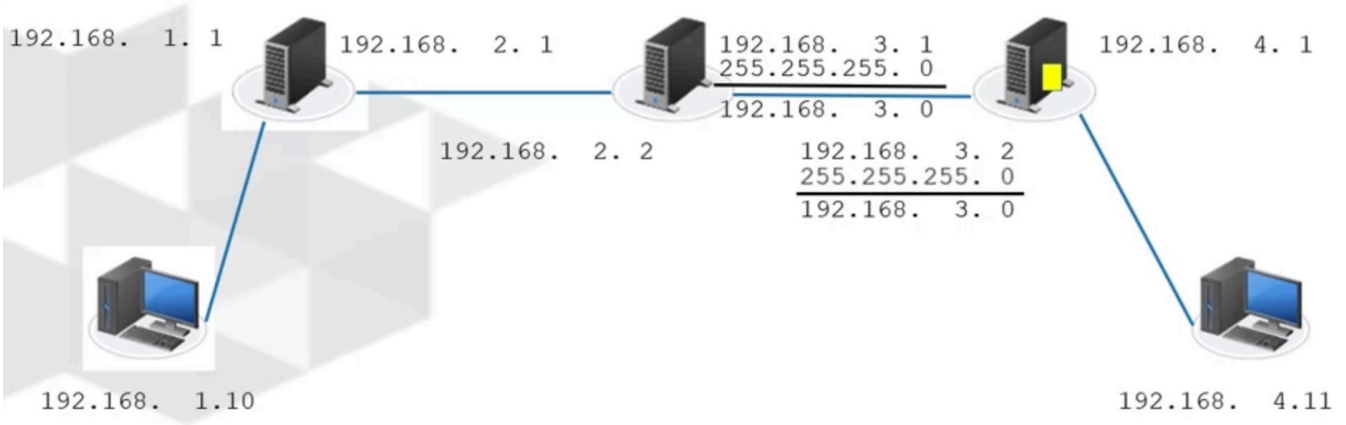
let me check the default gateway. It uses its subnet mask to do the ending for the default gateway. It comes up with the default gateway is on 1921681.0 network. Oh great, that's my network we can talk. It'll do it for the Mac address of the default gateway, and then it sends the data to the default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



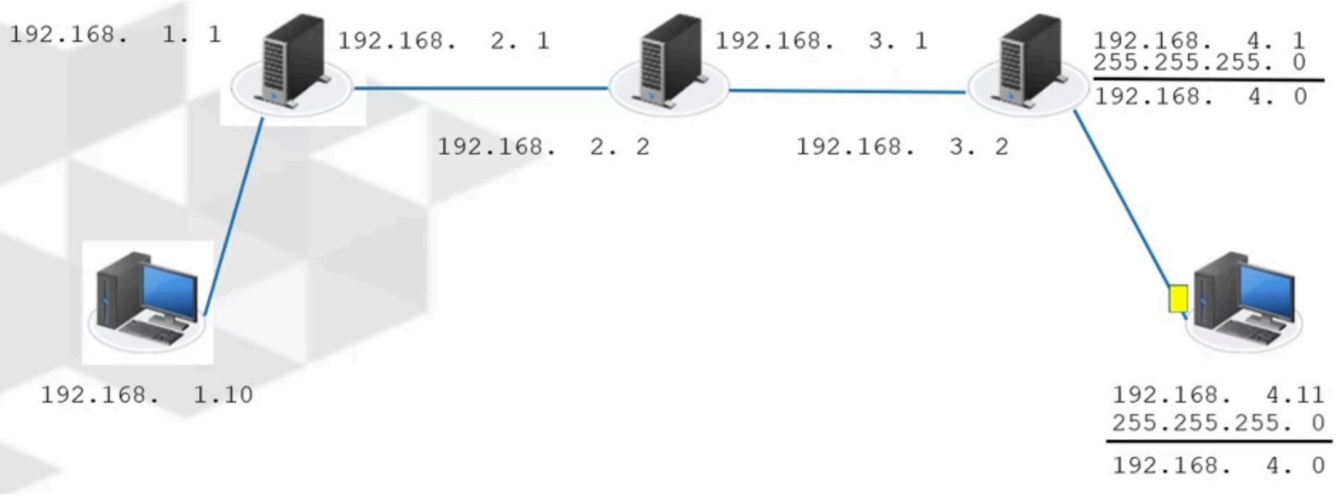
Now that first router gets it and it says, well, gee, I'm going to look at my routing table. My routing table says I should send it to that router in the middle. I have a network card 1921682.1 that happens to have a subnet mask of 255255255.0. What network am I on? Well, I'm on the 1921682.0 network. 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

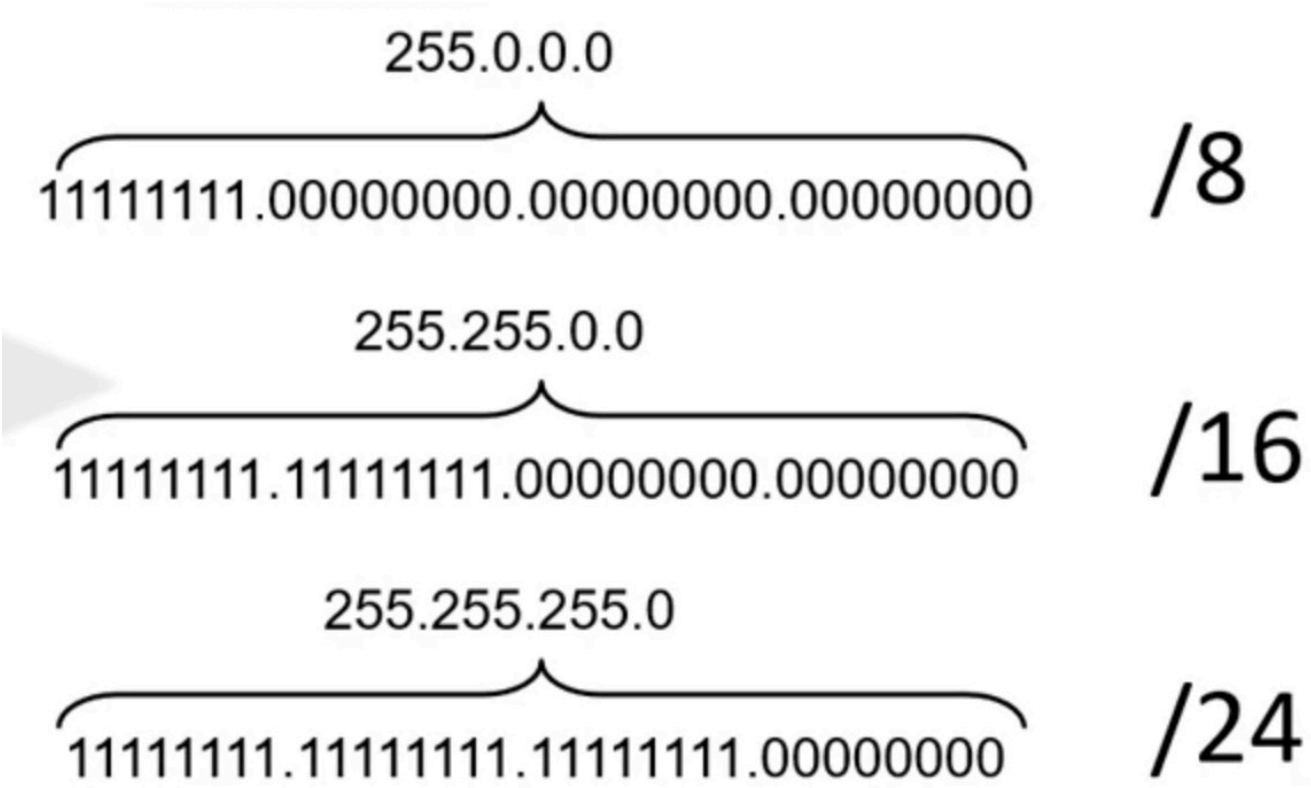
All CIDR notation is is a way of expressing the subnet mask as a slash number. We know that a binary number with eight ones is 255 in decimal. We know what the subnet mask, the ones are on the left, the zeros are on the right. We don't mix them up. At some point it stops being ones, it starts being zero.

- 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.

255.255.255.0

11111111.11111111.11111111.00000000

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.

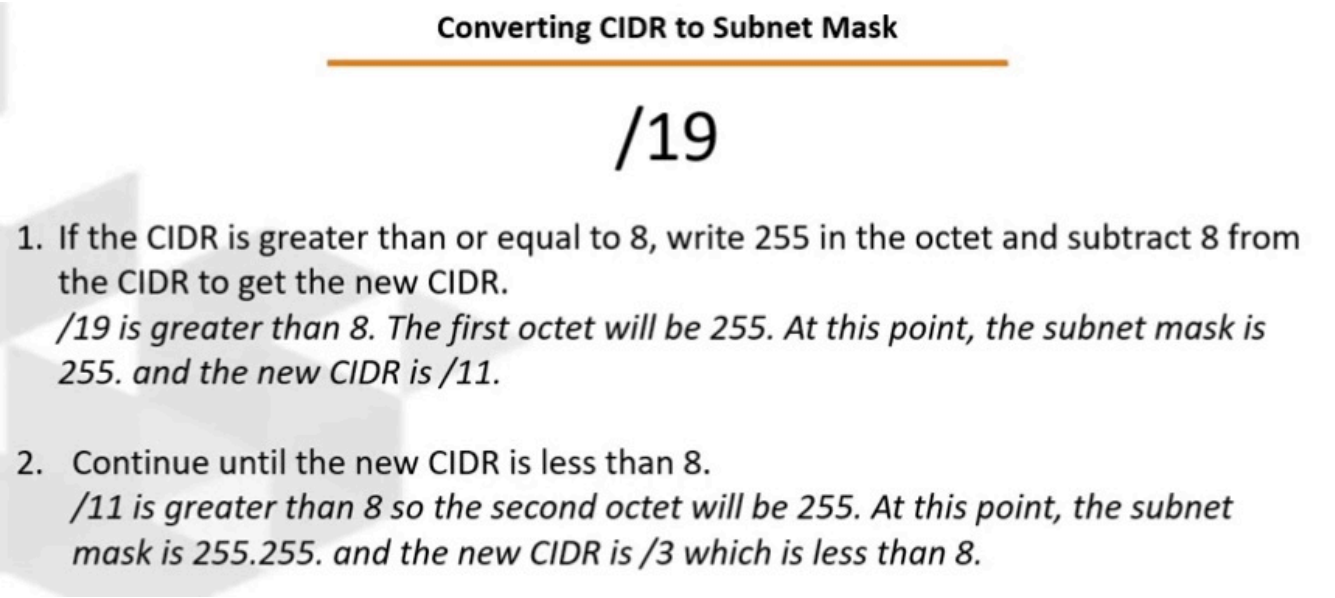


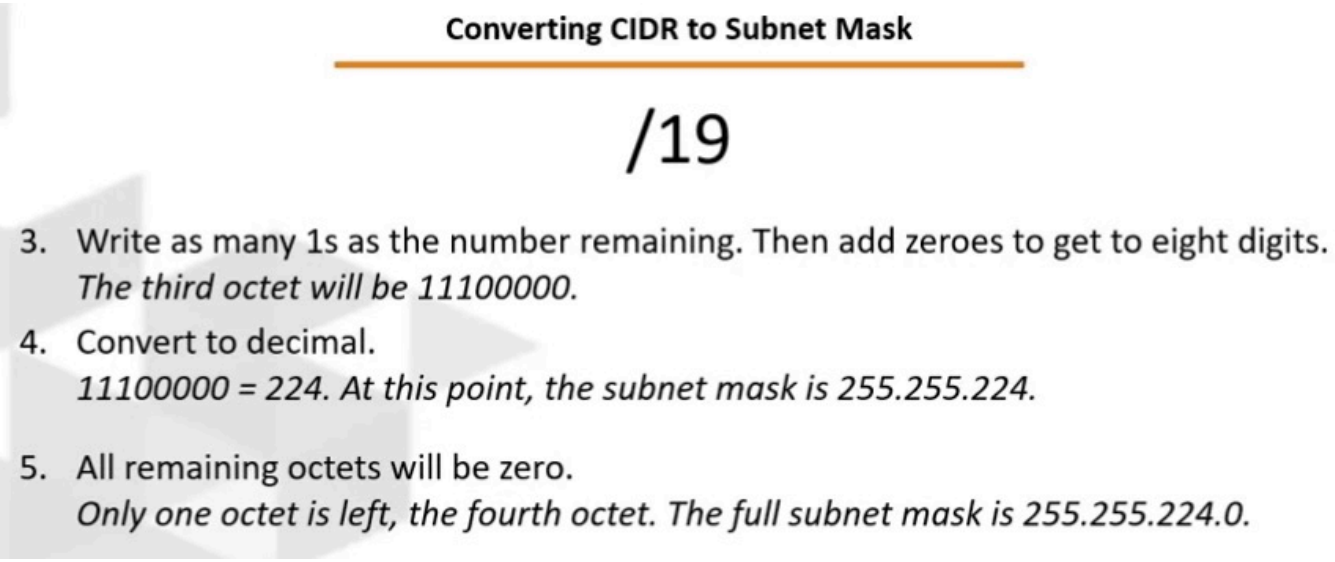
Now you might see a CIDR notation for a subnetted network. That means that that number is not /8, /16 or /24. How can we convert that to a subnet mask. E.g. what about a /19? Well, we could just write it out. You can just write out 19 ones and all the zeros and then convert each of those octets from 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

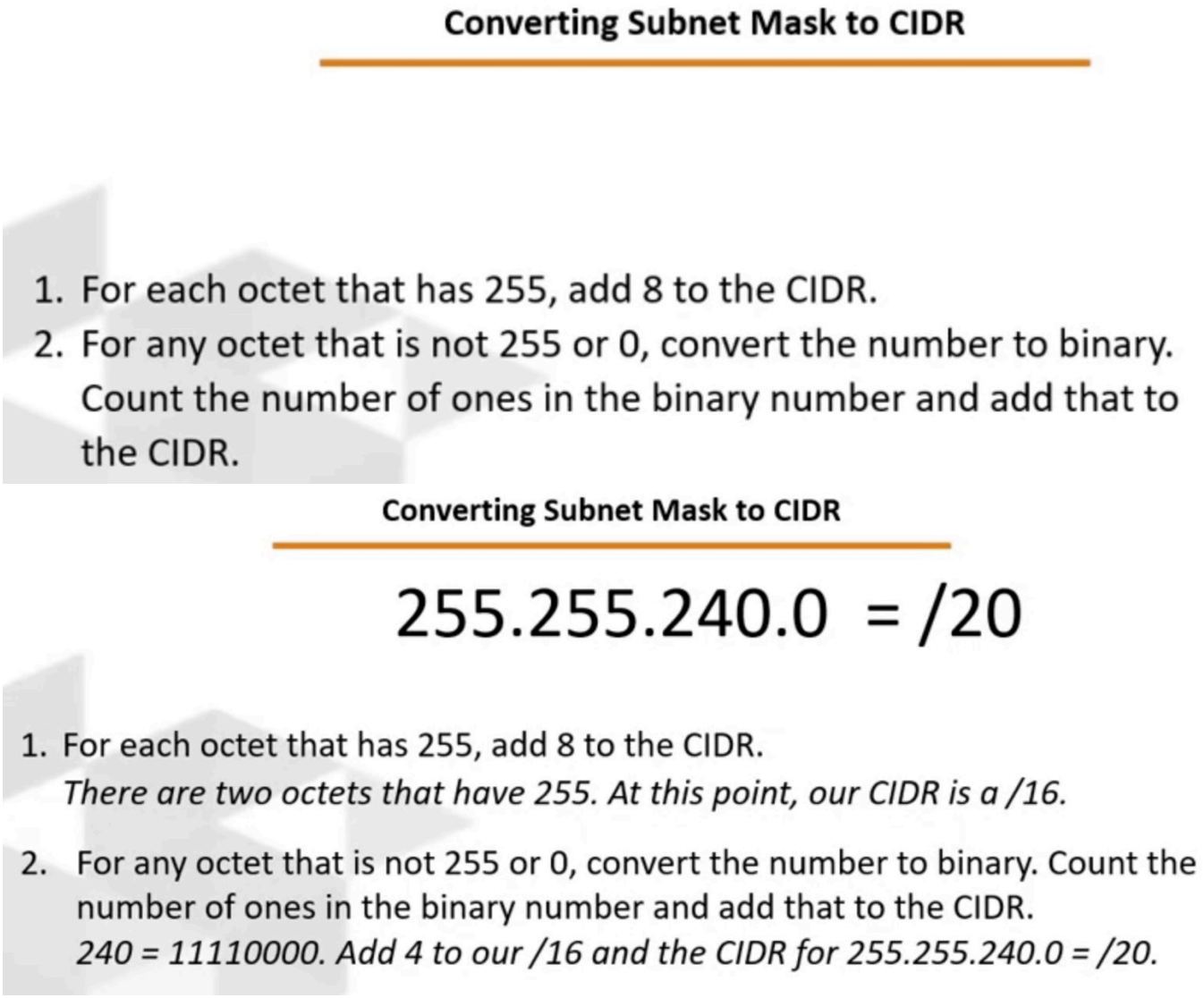
that /19. Rule Number 1 or step Number 1 says, well, if your CIDR is greater than or equal to eight, you write 255 in the octet and you subtract eight. Nineteen is greater than eight, so our first octet is going to be 255 and then our new CIDR is going to be 11. Second step says, well, we're going to 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.

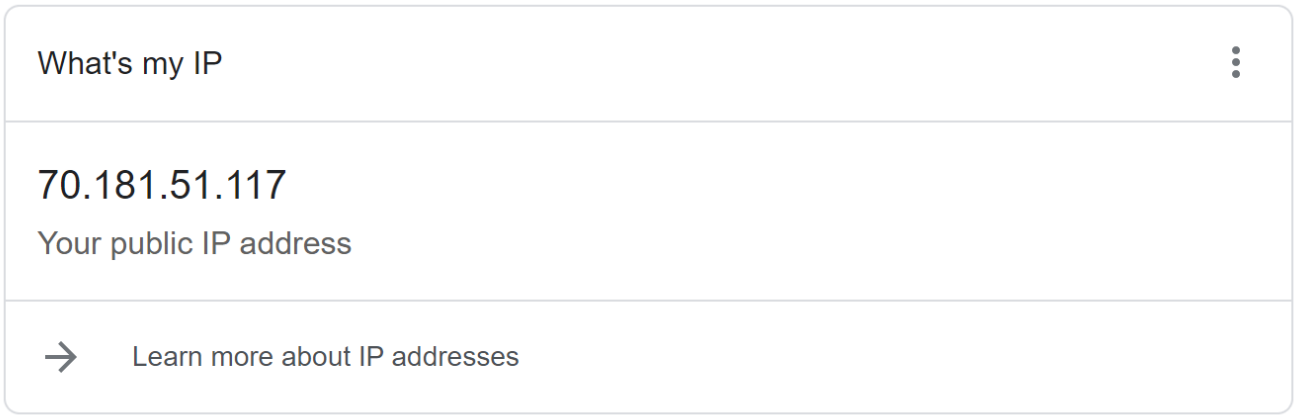
Network Address Translation

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

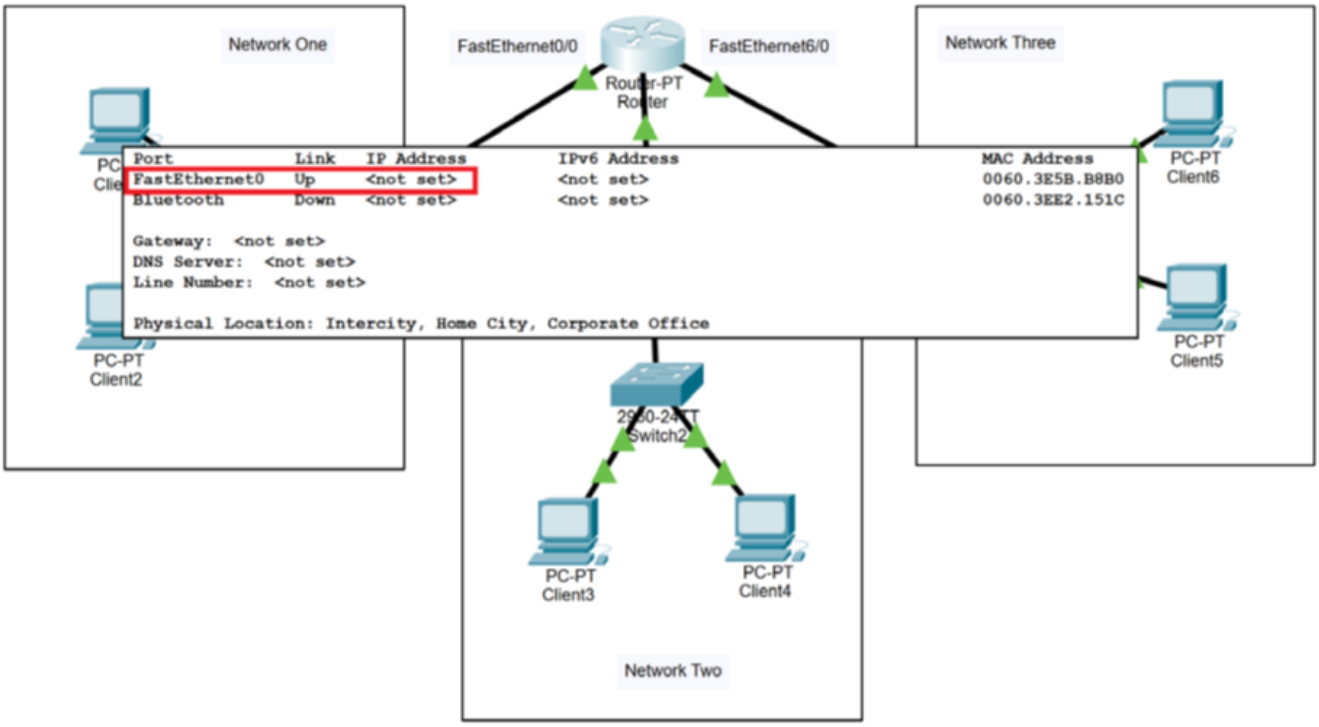
In this lab, you will design and implement an IPv4 network. Note: unlike previous labs, this lab will not provide a lot of screenshots. If you cannot remember how to do something, refer to the pictures in the previous labs.
There are several correct answers to this lab. The steps in the lab are provided 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

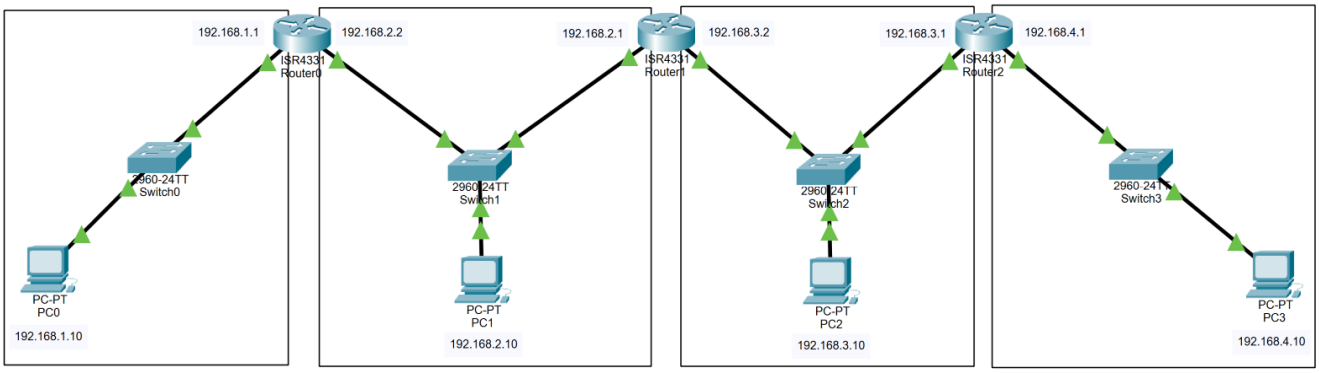
In this task, you will test your configuration.
1. Click on **Client1** to open the **Client1 Properties** dialog box.
2. In the **Desktop** tab, click the **Command Prompt** icon.
3. In the **Command Prompt**, use the **ping** command to verify connectivity to all the other computers in the network.
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

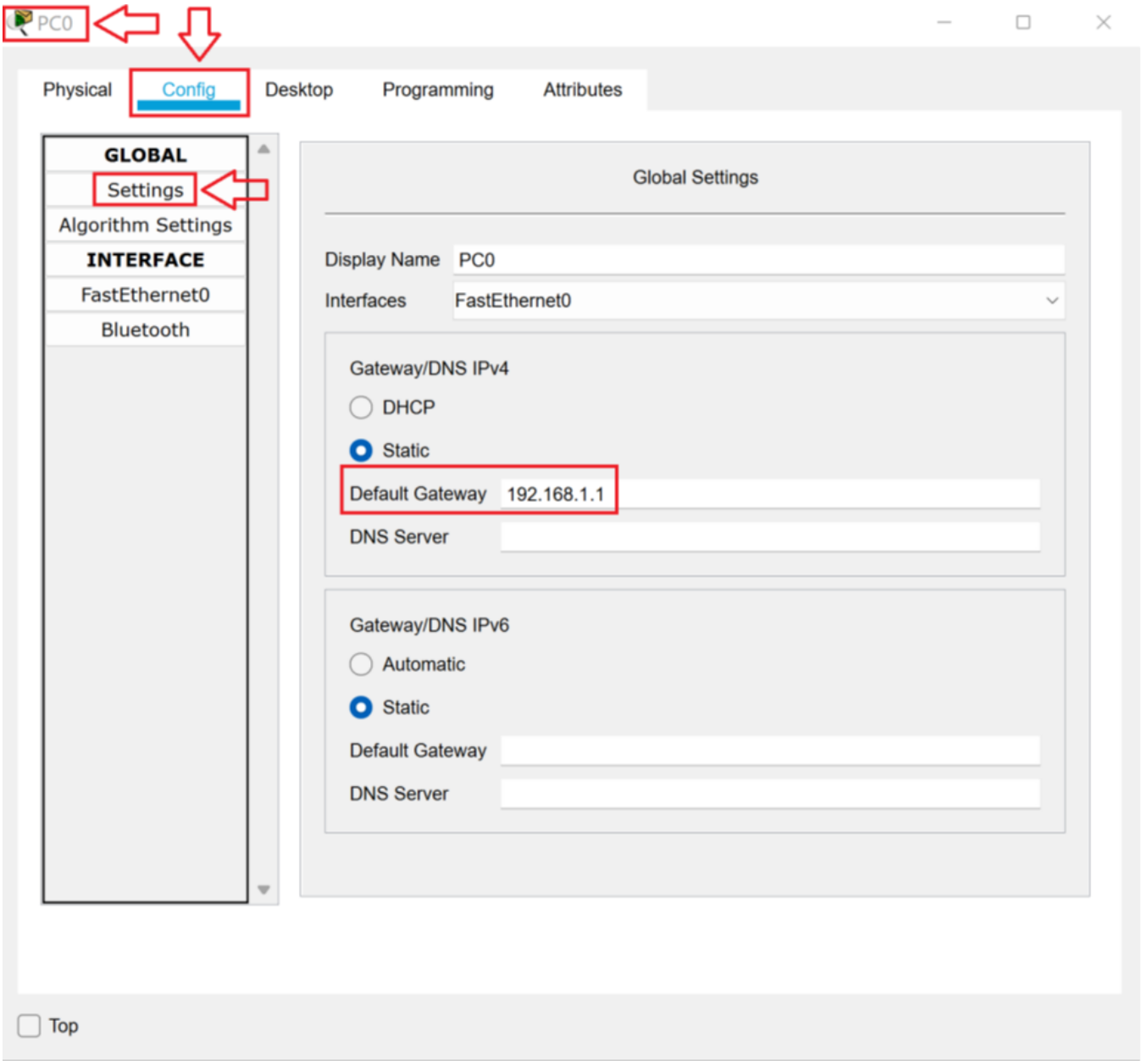
n this lab, you will configure routing. The lab is complete when all four of the clients can ping each other by IP address.

TASK A

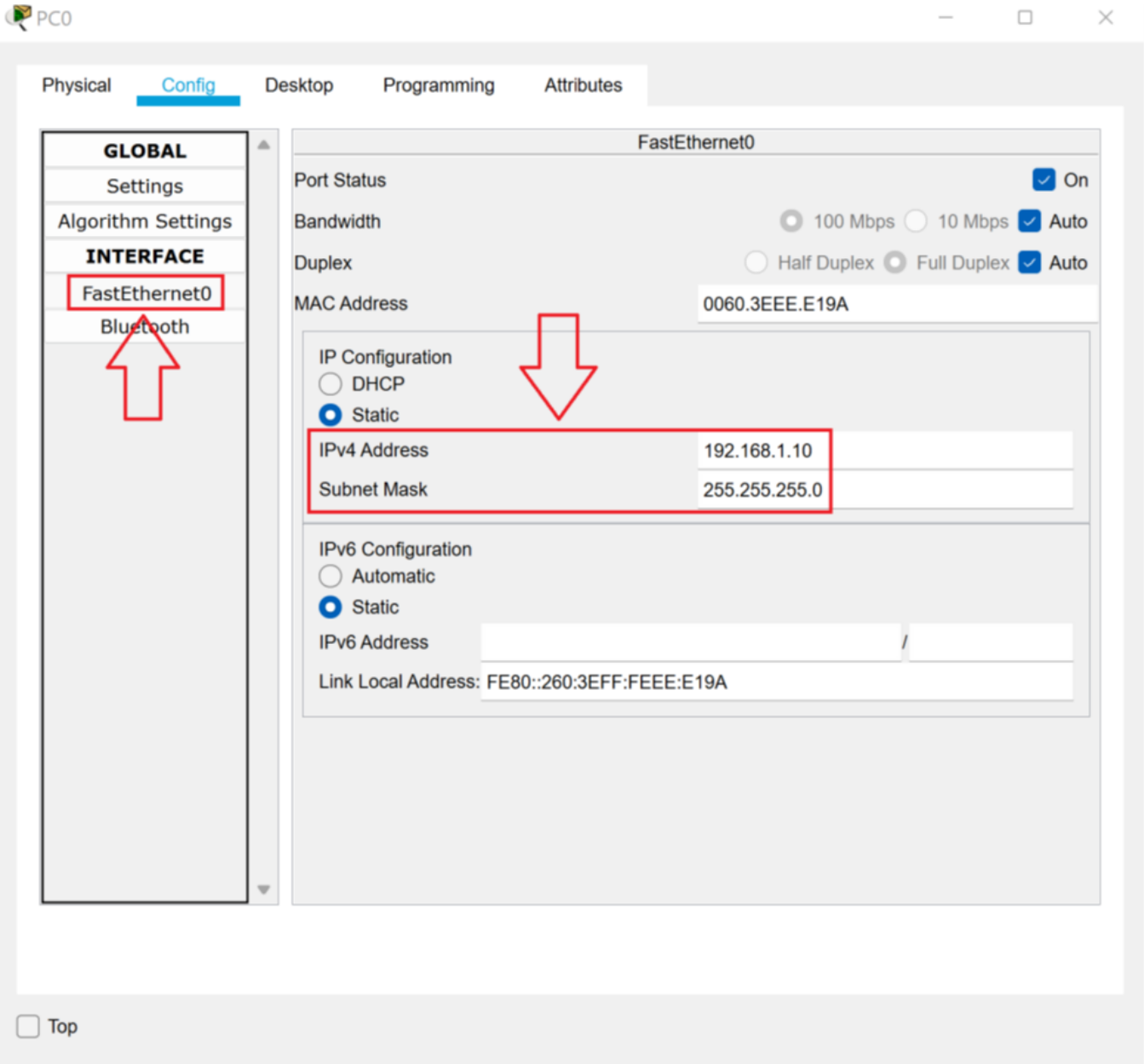
In this task, you examine the network setup.
1. Download the **4.4.2 Lab File** and open it in **Packet Tracer**.
4.4.2 Lab File
PKT File



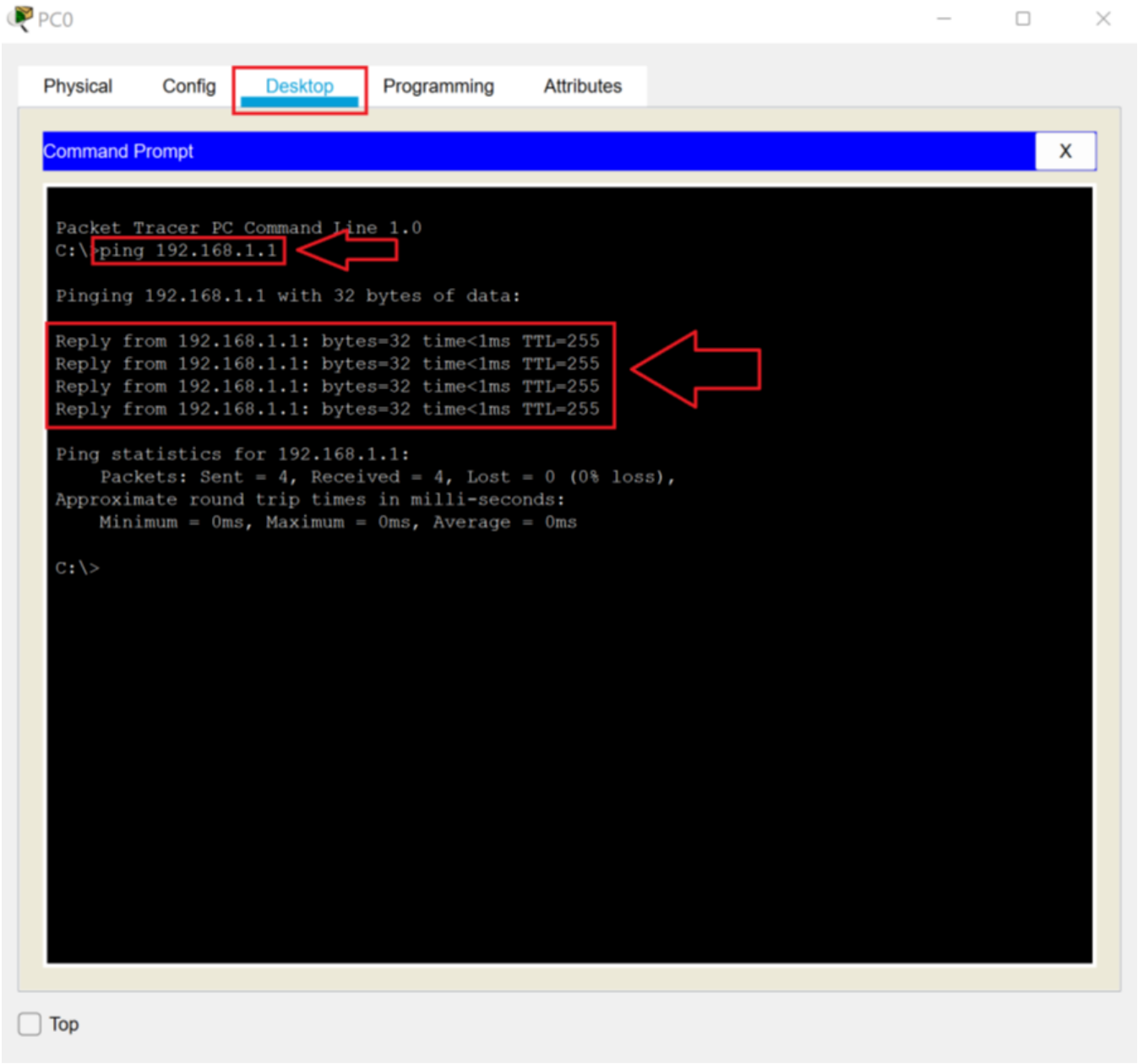
2. Click **PC0** to open the **PC0 Properties** dialog box. On the **Config** tab, in the **Global Settings** menu, observe the **Default Gateway**.



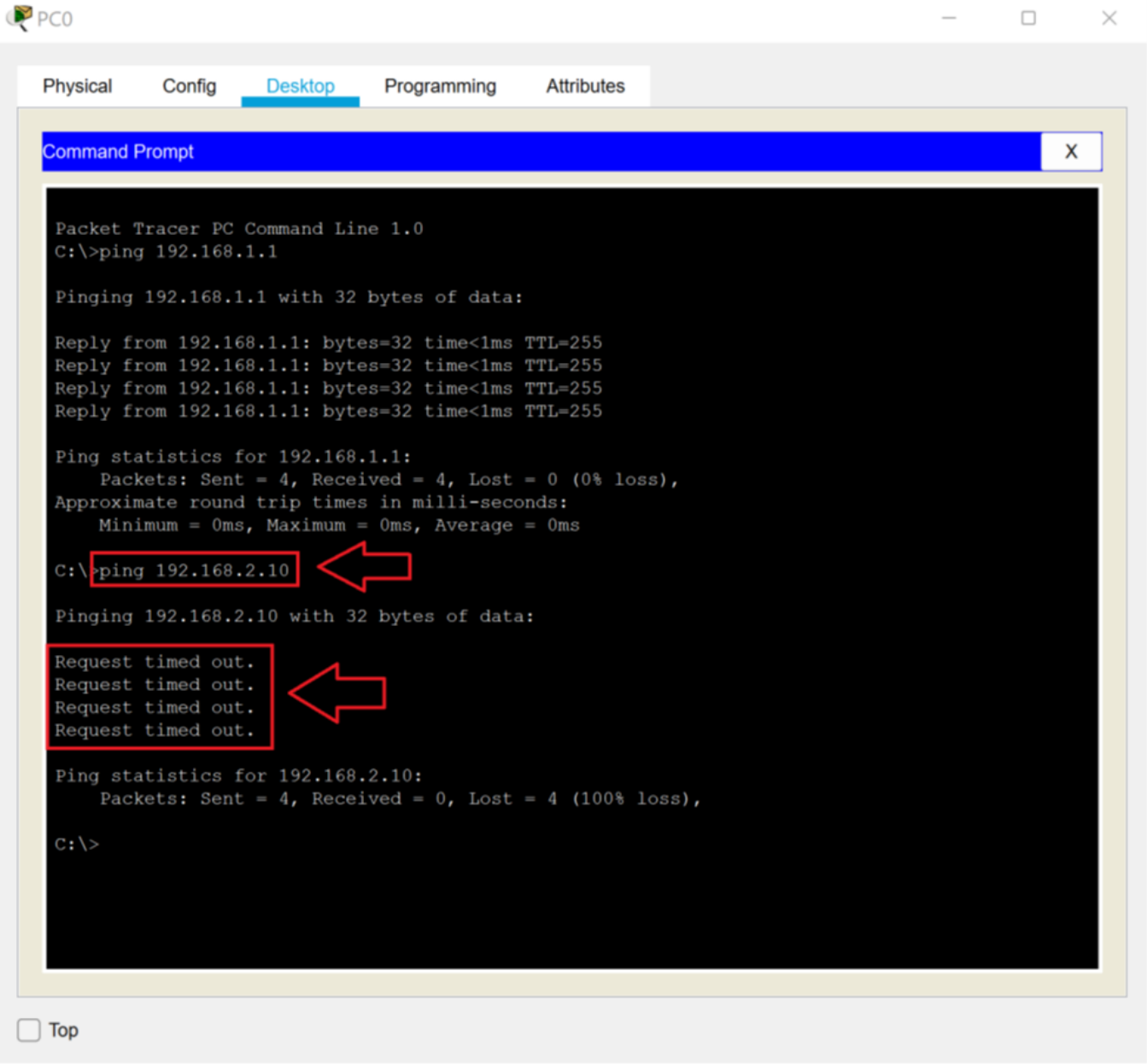
3. Click the **FastEthernet0** menu. Observe the client **IPv4 Address** and **Subnet Mask**.



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.



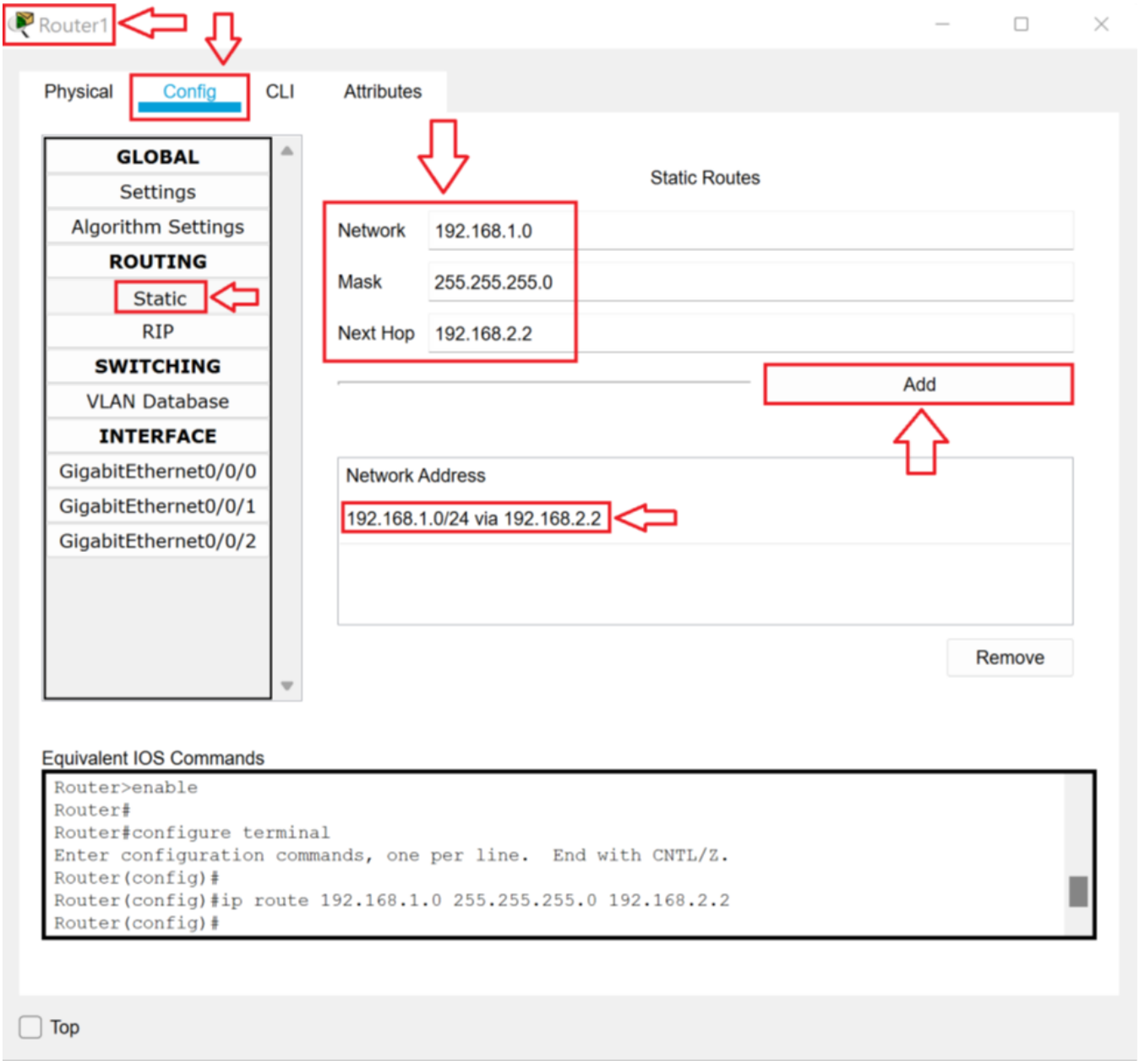
6. In the **Command Prompt** use the **ping** command to ping **192.168.3.10** and **192.168.4.10**. Notice the Destination Host Unreachable message.
7. Close the **PC0 Properties** dialog box.
8. Follow the same procedure to examine the Default Gateway, IPv4 Address and Subnet Masks of the other three 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

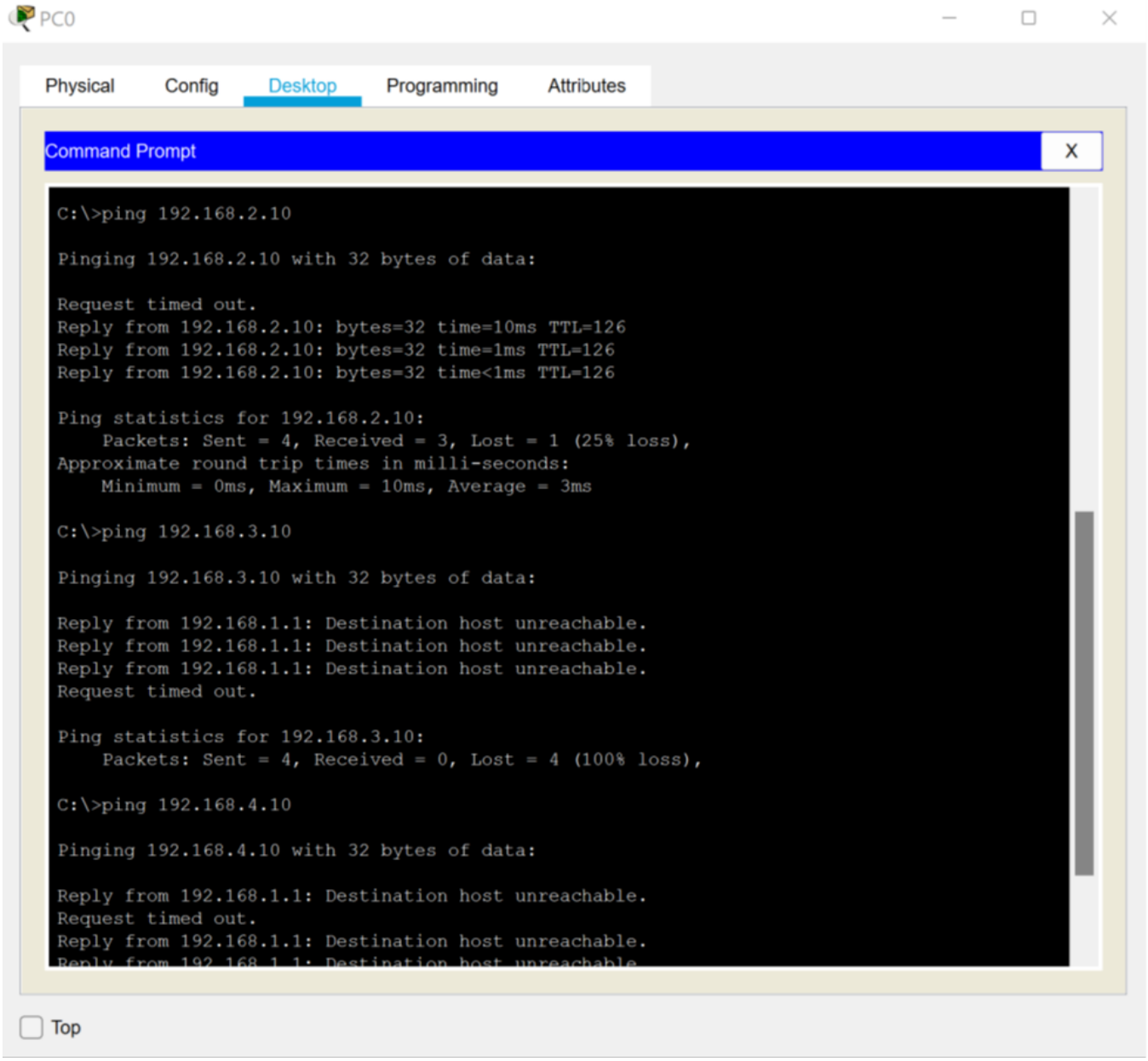
In this task, will resolve the routing problem between the 192.168.1.0 /24 network and the 192.168.2.0 /24 network. When PC0 pings PC1, PC0 sends the ping to Router0 at 192.168.1.1. Router0 sends the ping out its 192.168.2.2 interface. The ping arrives at PC1. PC1 sends the reply to Router1 at 192.168.2.1. 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.

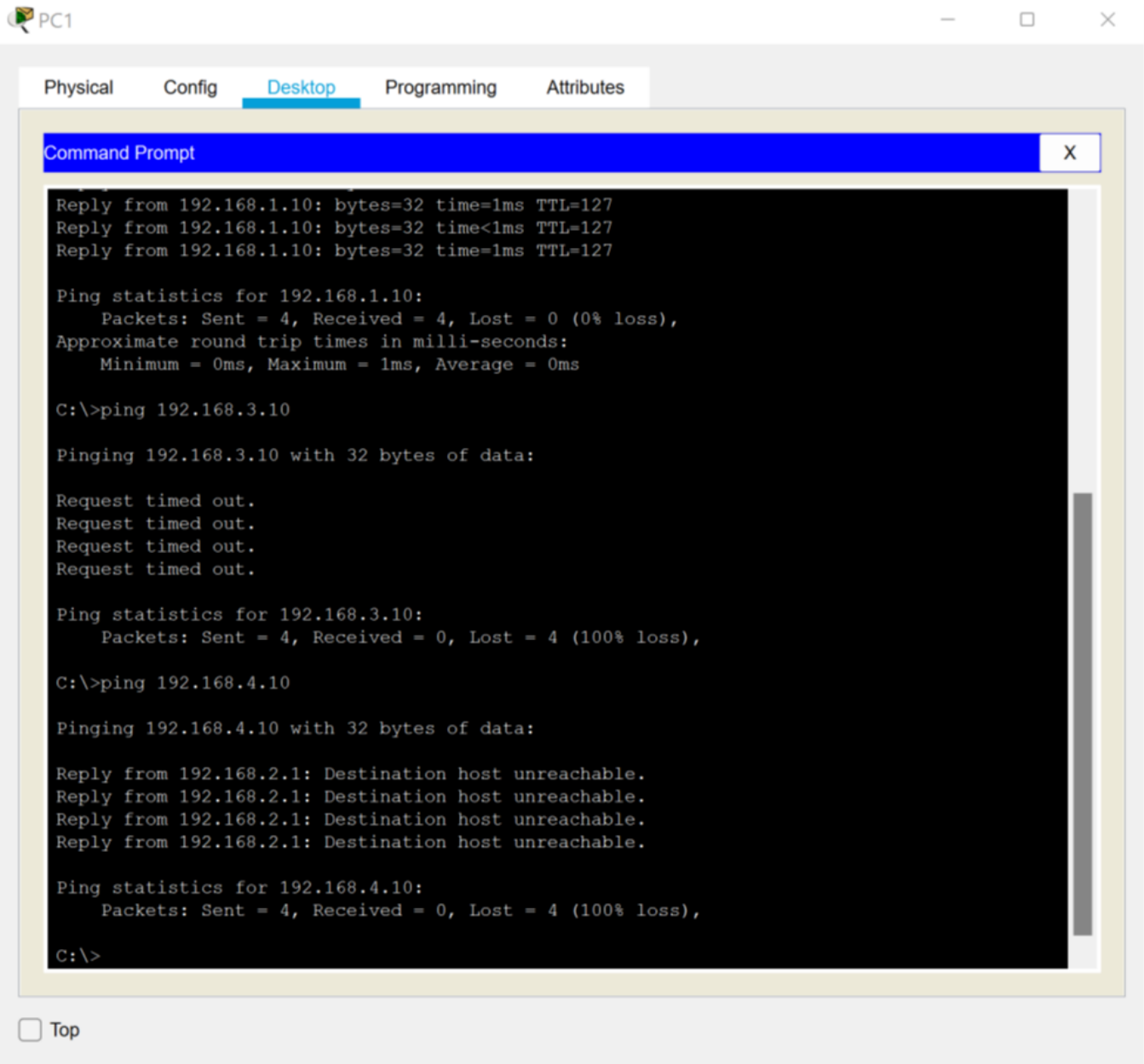
1. Click **Router1** to open the **Router1 Properties** dialog box. On the **Config** tab, click **Static**. In the **Network** text box, type **192.168.1.0**. In the **Mask** text box, type **255.255.255.0**. In the **Next Hop** text box, type **192.168.2.2**. Click the **Add** button. Notice the route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



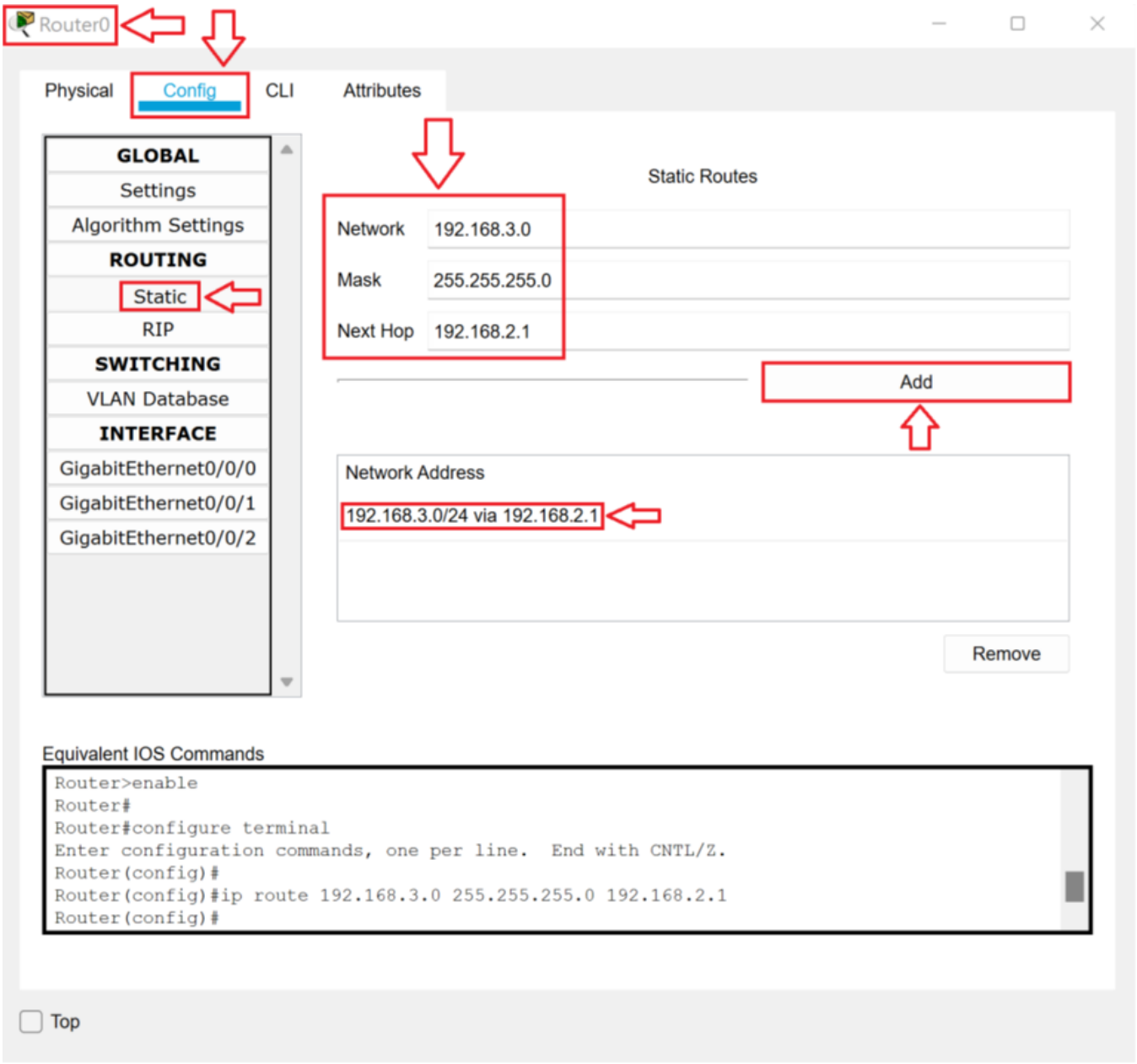
3. On PC1, verify PC1 can now ping PC0 but it still cannot ping PC2 or PC3.



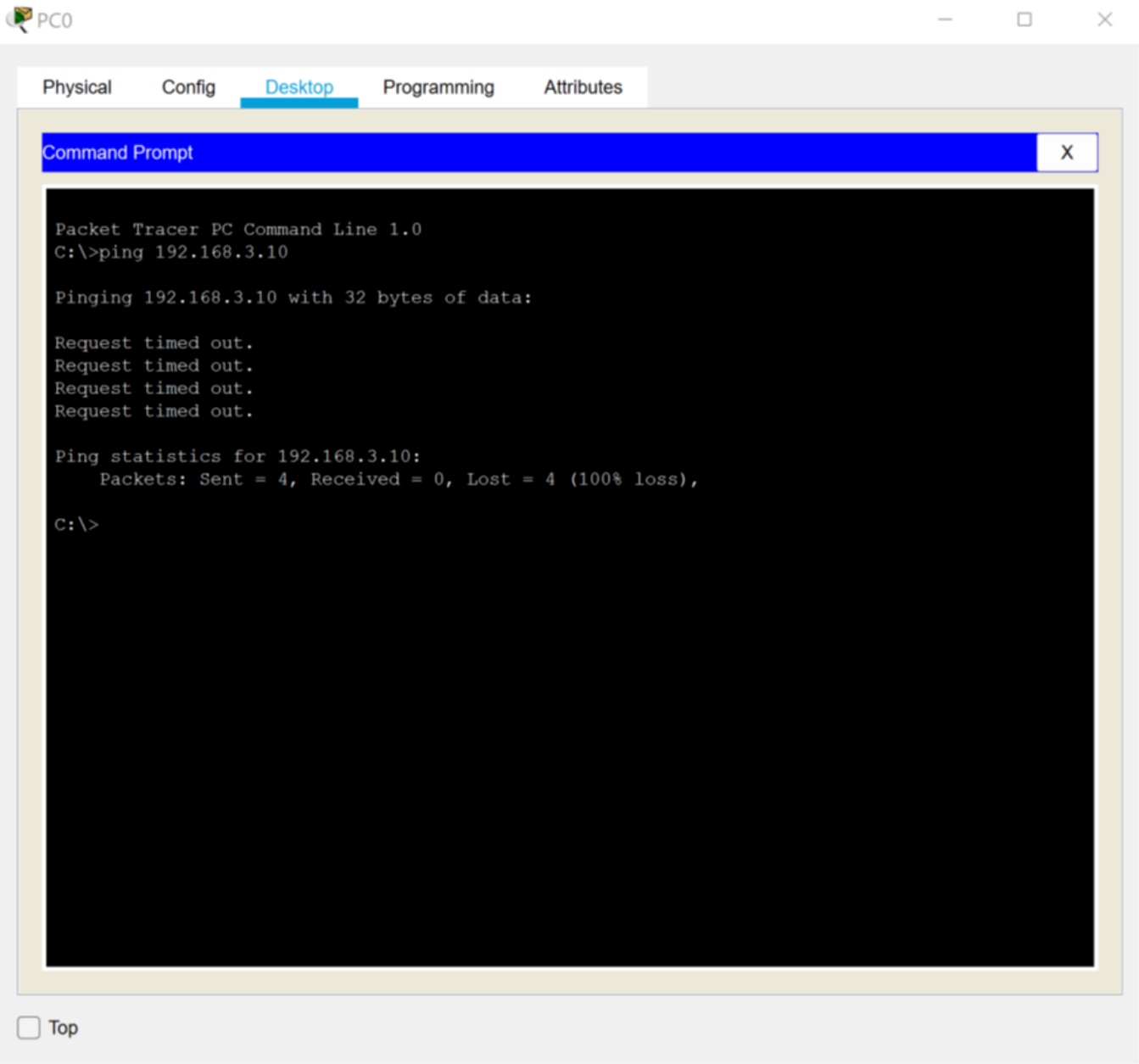
TASK C

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.

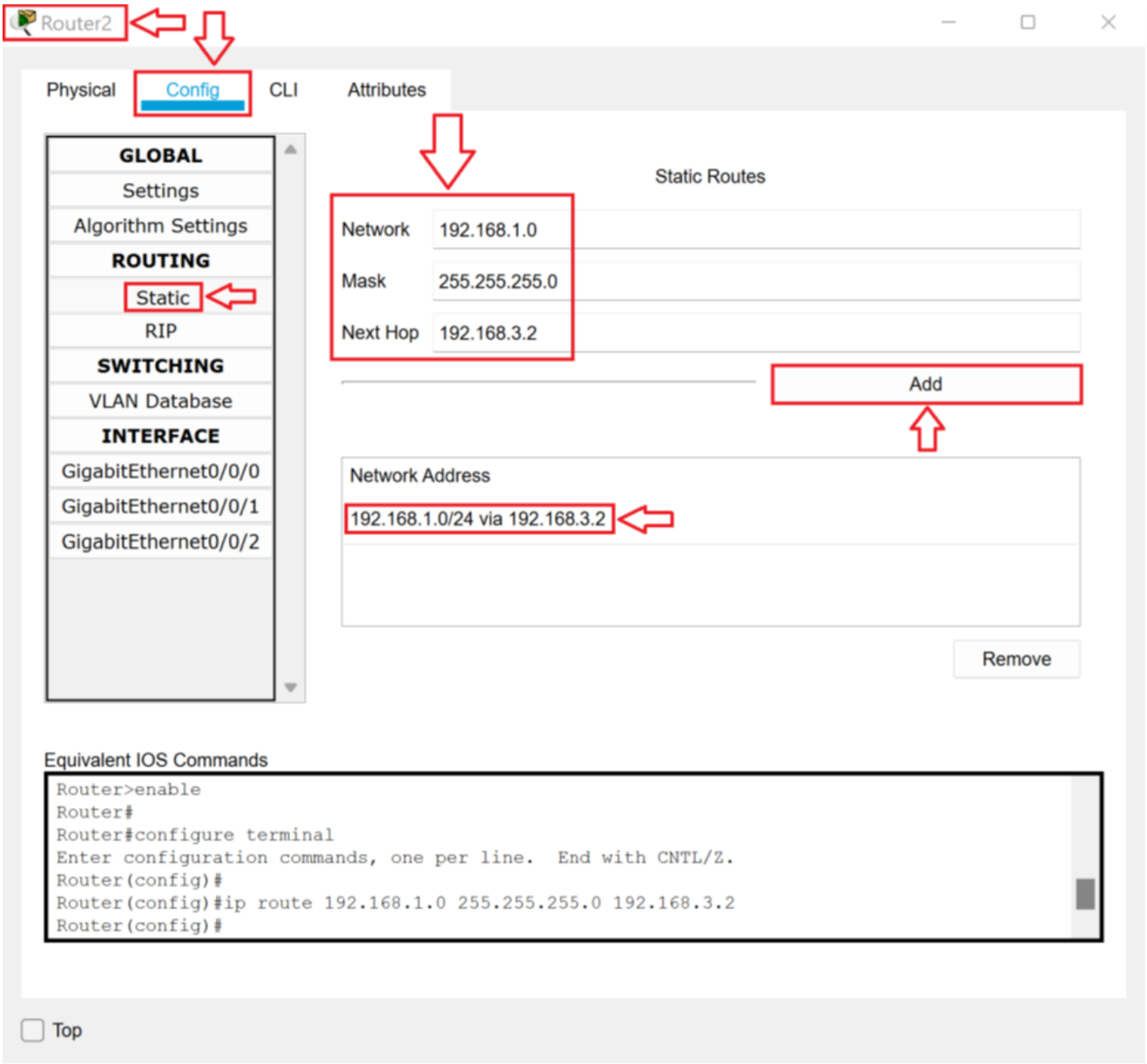
1. Click **Router0** to open the **Router0 Properties** dialog box. On the **Config** tab, click **Static**. In the **Network** text box, type **192.168.3.0**. In the **Mask** text box, type **255.255.255.0**. In the **Next Hop** text box, type **192.168.2.1**. Click the **Add** button. Notice the route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



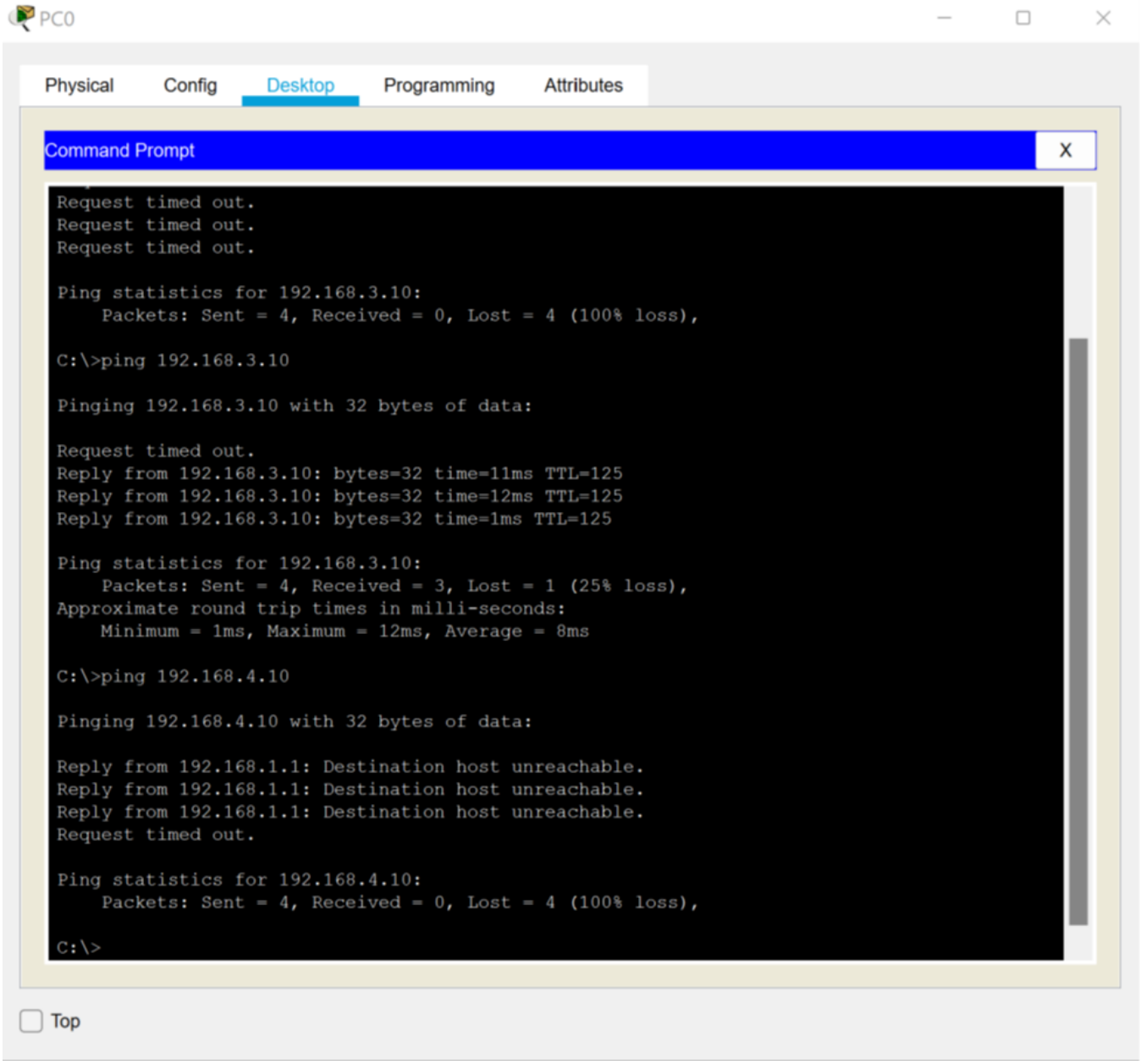
2. On **PC0**, verify that the client no longer gets a Destination Host Unreachable error. Now it gets a Request Timed Out. When **PC0** pings **PC2**, **PC0** sends the ping to **Router0** at 192.168.1.1. Now **Router0** sends the data to **Router1** at 192.168.2.1. **Router1** has an interface on the 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.



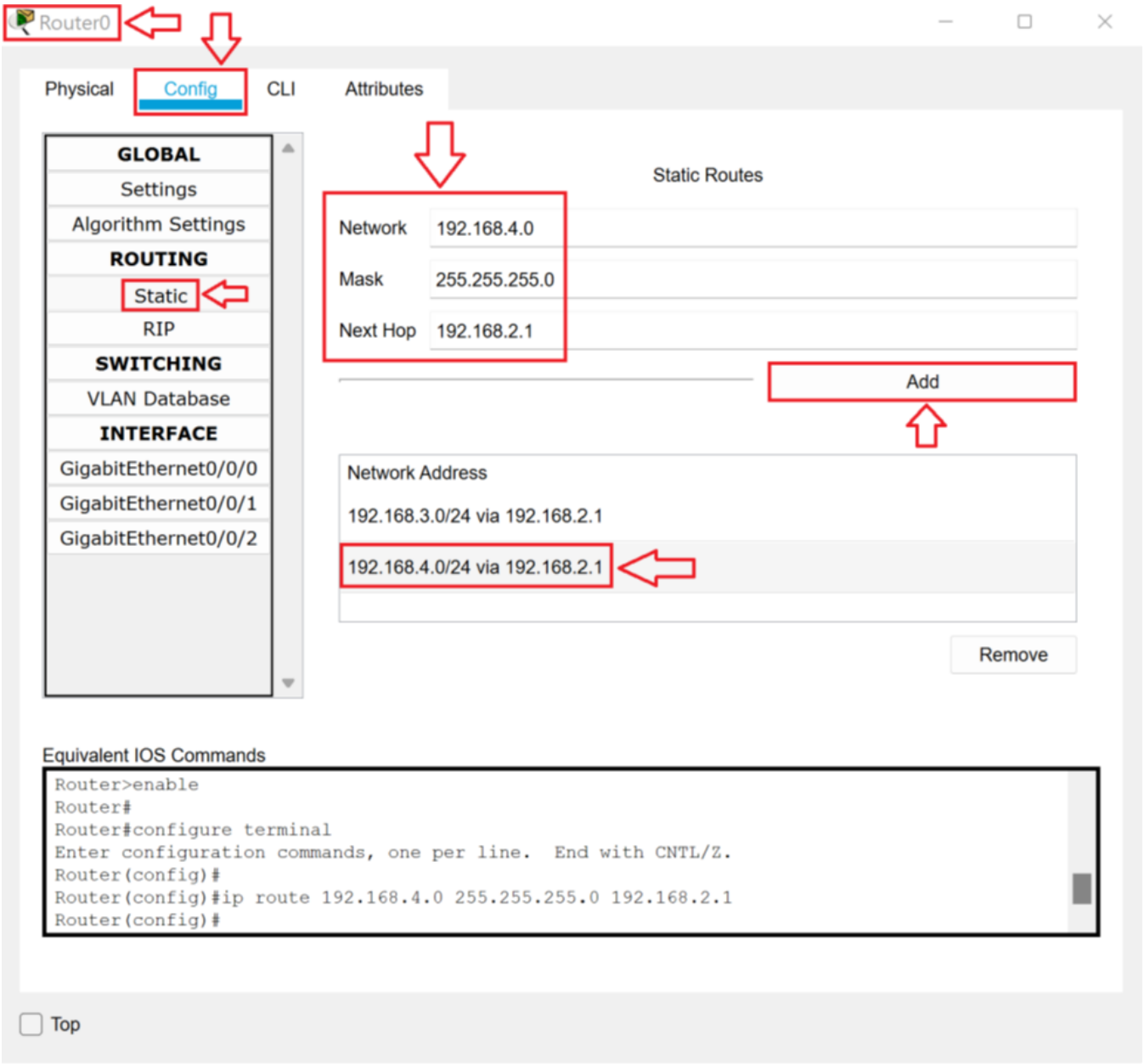
4. Verify that **PC0** can now ping **PC2**. It still gets a Destination Host Unreachable reply when pinging **PC3**.



TASK D

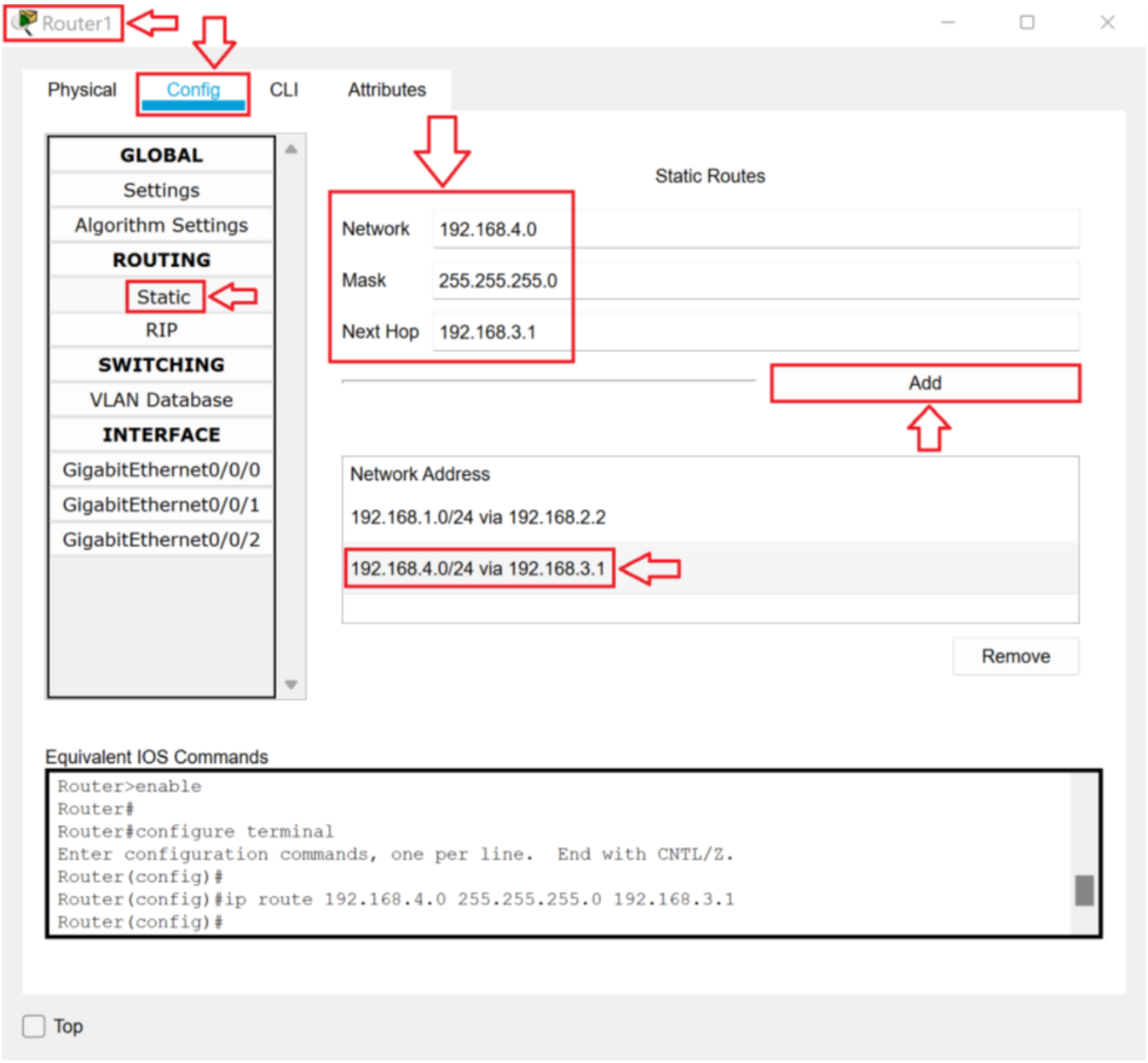
In this task, will resolve the routing problem between the 192.168.1.0 /24 network and the 192.168.4.0 /24 network. When PC0 pings PC3, PC0 sends the ping to Router0 at 192.168.1.1. Router0 does not have an interface in the 192.168.4.0 /24 or an entry in its routing table. You must start by making an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