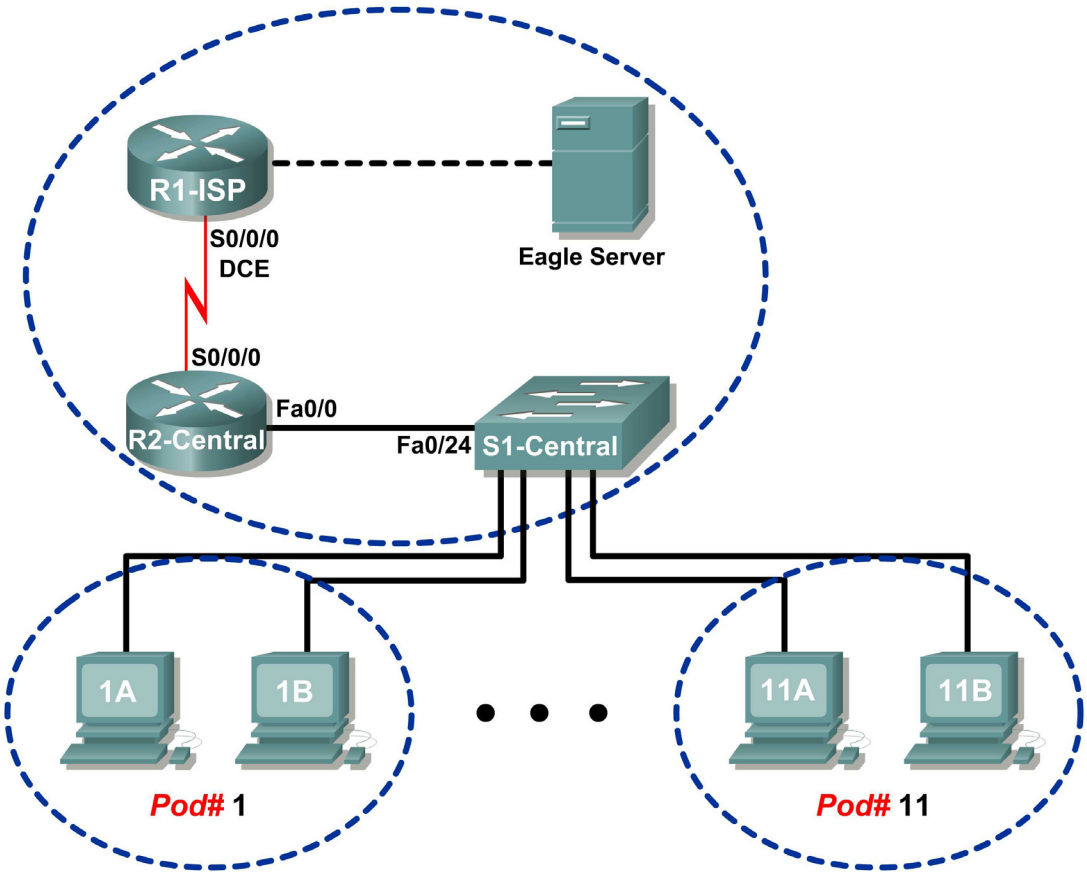
Lab 6.7.1: Ping and Traceroute

Topology Diagram



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1-ISP	S0/0/0	10.10.10.6	255.255.255.252	N/A
	Fa0/0	192.168.254.253	255.255.255.0	N/A
R2-Central	S0/0/0	10.10.10.5	255.255.255.252	N/A
	Fa0/0	172.16.255.254	255.255.0.0	N/A
Eagle Server	N/A	192.168.254.254	255.255.255.0	192.168.254.253
	N/A	172.31.24.254	255.255.255.0	N/A
host*Pod#*A	N/A	172.16.*Pod#*.1	255.255.0.0	172.16.255.254
host*Pod#*B	N/A	172.16.*Pod#*.2	255.255.0.0	172.16.255.254
S1-Central	N/A	172.16.254.1	255.255.0.0	172.16.255.254

Learning Objectives

Upon completion of this lab, you will be able to:

- Use the **ping** command to verify simple TCP/IP network connectivity.
- Use the **tracert/traceroute** command to verify TCP/IP connectivity.

Background

Two tools that are indispensable when testing TCP/IP network connectivity are **ping** and **tracert**. The **ping** utility is available on Windows, Linux, and Cisco IOS, and tests network connectivity. The **tracert** utility is available on Windows, and a similar utility, **traceroute**, is available on Linux and Cisco IOS. In addition to testing for connectivity, **tracert** can be used to check for network latency.

For example, when a web browser fails to connect to a web server, the problem can be anywhere between client and the server. A network engineer may use the **ping** command to test for local network connectivity or connections where there are few devices. In a complex network, the **tracert** command would be used. Where to begin connectivity tests has been the subject of much debate; it usually depends on the experience of the network engineer and familiarity with the network.

The Internet Control Message Protocol (ICMP) is used by both **ping** and **tracert** to send messages between devices. ICMP is a TCP/IP Network layer protocol, first defined in RFC 792, September, 1981. ICMP message types were later expanded in RFC 1700.

Scenario

In this lab, the **ping** and **tracert** commands will be examined, and command options will be used to modify the command behavior. To familiarize the students with the use of the commands, devices in the Cisco lab will be tested.

Measured delay time will probably be less than those on a production network. This is because there is little network traffic in the Eagle 1 lab.

Task 1: Use the ping Command to Verify Simple TCP/IP Network Connectivity.

The **ping** command is used to verify TCP/IP Network layer connectivity on the local host computer or another device in the network. The command can be used with a destination IP address or qualified name, such as eagle-server.example.com, to test domain name services (DNS) functionality. For this lab, only IP addresses will be used.

The **ping** operation is straightforward. The source computer sends an ICMP echo request to the destination. The destination responds with an echo reply. If there is a break between the source and destination, a router may respond with an ICMP message that the host is unknown or the destination network is unknown.

Step 1: Verify TCP/IP Network layer connectivity on the local host computer.

```
C:\> ipconfig
Windows IP Configuration
Ethernet adapter Local Area Connection:
        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 172.16.1.2
        Subnet Mask . . . . . . . . . . . : 255.255.0.0
        Default Gateway . . . . . . . . . : 172.16.255.254
C:\>
```

Figure 1. Local TCP/IP Network Information

1. Open a Windows terminal and determine IP address of the pod host computer with the **ipconfig** command, as shown in Figure 1.

   The output should look the same except for the IP address. Each pod host computer should have the same network mask and default gateway address; only the IP address may differ. If the information is missing or if the subnet mask and default gateway are different, reconfigure the TCP/IP settings to match the settings for this pod host computer.

2. Record information about local TCP/IP network information:

TCP/IP Information	Value
IP Address	10.100.8.72
Subnet Mask	255.255.0.0
Default Gateway	172.16.255.254

```
C:\> ping 172.16.1.2❶
Pinging 172.16.1.1 with 32 bytes of data:
❷Reply from 172.16.1.2: bytes=32 time<1ms TTL=128
Reply from 172.16.1.2: bytes=32 time<1ms TTL=128
Reply from 172.16.1.2: bytes=32 time<1ms TTL=128
Reply from 172.16.1.2: bytes=32 time<1ms TTL=128

❸Ping statistics for 172.16.1.2:
    Packets: Sent❹ = 4, Received❺ = 4, Lost❻ = 0 (0% loss),
❼Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>
```

Figure 2. Output of the ping Command on the Local TCP/IP Stack

3. Use the **ping** command to verify TCP/IP Network layer connectivity on the local host computer.

   By default, four ping requests are sent to the destination and reply information is received. Output should look similar to that shown in Figure 2.

   ❶ Destination address, set to the IP address for the local computer.

   ❷ Reply information:

   bytes—size of the ICMP packet.

   time—elapsed time between transmission and reply.

   TTL—default TTL value of the DESTINATION device, minus the number of routers in the path. The maximum TTL value is 255, and for newer Windows machines the default value is 128.

   ❸ Summary information about the replies:

   ❹ Packets Sent—number of packets transmitted. By default, four packets are sent.

   ❺ Packets Received—number of packets received.

   ❻ Packets Lost —difference between number of packets sent and received.

   ❼ Information about the delay in replies, measured in milliseconds. Lower round trip times indicate faster links. A computer timer is set to 10 milliseconds. Values faster than 10 milliseconds will display 0.

4. Fill in the results of the **ping** command on your computer:

Field	Value
Size of packet	32 bytes
Number of packets sent	4
Number of replies	4
Number of lost packets	0
Minimum delay	0ms
Maximum delay	0ms
Average delay	0ms

Step 2: Verify TCP/IP Network layer connectivity on the LAN.

```
C:\> ping 172.16.255.254
Pinging 172.16.255.254 with 32 bytes of data:
Reply from 172.16.255.254: bytes=32 time=1ms TTL=255
Reply from 172.16.255.254: bytes=32 time<1ms TTL=255
Reply from 172.16.255.254: bytes=32 time<1ms TTL=255
Reply from 172.16.255.254: bytes=32 time<1ms TTL=255
Ping statistics for 172.16.255.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
C:\>
```

Figure 3. Output of the ping Command to the Default Gateway

1. Use the **ping** command to verify TCP/IP Network layer connectivity to the default gateway. Results should be similar to those shown in Figure 3.

   Cisco IOS default TTL value is set to 255. Because the datagrams did not travel through a router, the TTL value returned is 255.

2. Fill in the results of the **ping** command to the default Gateway:

Field	Value
Size of packet	32 bytes
Number of packets sent	4
Number of replies	4
Number of lost packets	0ms
Minimum delay	0ms
Maximum delay	1ms
Average delay	0ms

What would be the result of a loss of connectivity to the default gateway?

could not reach external networks

Step 3: Verify TCP/IP Network layer connectivity to a remote network.

```
C:\> ping 192.168.254.254
Pinging 192.168.254.254 with 32 bytes of data:
Reply from 192.168.254.254: bytes=32 time<1ms TTL=62
Reply from 192.168.254.254: bytes=32 time<1ms TTL=62
Reply from 192.168.254.254: bytes=32 time<1ms TTL=62
Reply from 192.168.254.254: bytes=32 time<1ms TTL=62
Ping statistics for 192.168.254.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>
```

Figure 4. Output of the `ping` Command to Eagle Server

1. Use the **`ping`** command to verify TCP/IP Network layer connectivity to a device on a remote network. In this case, Eagle Server will be used. Results should be similar to those shown in Figure 4.

   Linux default TTL value is set to 64. Since the datagrams traveled through two routers to reach Eagle Server, the returned TTL value is 62.

2. Fill in the results of the **`ping`** command on your computer:

Field	Value
Size of packet	32 bytes
Number of packets sent	4
Number of replies	4
Number of lost packets	0
Minimum delay	0ms
Maximum delay	0ms
Average delay	0ms

```
C:\ > ping 192.168.254.254
Pinging 192.168.254.254 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 192.168.254.254:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\>
```

Figure 5. Output of a `ping` Command with Lost Packets

The **`ping`** command is extremely useful when troubleshooting network connectivity. However, there are limitations. In Figure 5, the output shows that a user cannot reach Eagle Server. Is the problem with Eagle Server or a device in the path? The **`tracert`** command, examined next, can display network latency and path information.

Task 2: Use the `tracert` Command to Verify TCP/IP Connectivity.

The **`tracert`** command is useful for learning about network latency and path information. Instead of using the **`ping`** command to test connectivity of each device to the destination, one by one, the **`tracert`** command can be used.

On Linux and Cisco IOS devices, the equivalent command is **`traceroute`**.

Step 1: Verify TCP/IP Network layer connectivity with the `tracert` command.

1. Open a Windows terminal and issue the following command:

```
C:\> tracert 192.168.254.254
```

```
C:\> tracert 192.168.254.254
Tracing route to 192.168.254.254 over a maximum of 30 hops
  1    <1 ms    <1 ms    <1 ms  172.16.255.254
  2    <1 ms    <1 ms    <1 ms  10.10.10.6
  3    <1 ms    <1 ms    <1 ms  192.168.254.254
Trace complete.
C:\>
```

Figure 6. Output of the `tracrt` command to Eagle Server.

Output from the **`tracert`** command should be similar to that shown in Figure 6.

2. Record your result in the following table:

Field	Value
Maximum number of hops	30
First router IP address	172.16.255.254
Second router IP address	10.10.10.6
Destination reached?	yes

Step 2: Observe `tracert` output to a host that lost network connectivity.

If there is a loss of connectivity to an end device such as Eagle Server, the **`tracert`** command can give valuable clues as to the source of the problem. The **`ping`** command would show the failure but not any other kind of information about the devices in the path. Referring to the Eagle 1 lab Topology Diagram, both R2-Central and R1-ISP are used for connectivity between the pod host computers and Eagle Server.

```
C:\> tracert -w 5 -h 4 192.168.254.254
Tracing route to 192.168.254.254 over a maximum of 4 hops
  1    <1 ms    <1 ms    <1 ms  172.16.255.254
  2    <1 ms    <1 ms    <1 ms  10.10.10.6
  3     *        *        *     Request timed out.
  4     *        *        *     Request timed out.

Trace complete.
C:\>
```

Figure 7. Output of the `tracert` Command

Refer to Figure 7. Options are used with the **`tracert`** command to reduce wait time (in milliseconds), **`-w 5`**, and maximum hop count, **`-h 4`**. If Eagle Server was disconnected from the network, the default gateway would respond correctly, as well as R1-ISP. The problem must be on the `192.168.254.0/24` network. In this example, Eagle Server has been turned off.

What would the **tracert** output be if R1-ISP failed?

nothing reached after r2 central

What would the **tracert** output be if R2-Central failed?

no connectivity

Task 3: Challenge

The default values for the **ping** command normally work for most troubleshooting scenarios. There are times, however, when fine tuning **ping** options may be useful. Issuing the **ping** command without any destination address will display the options shown in Figure 8:

```
C:\> ping

Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
            [-r count] [-s count] [[-j host-list] | [-k host-list]]
            [-w timeout] target_name

Options:
    -t              Ping the specified host until stopped.
                    To see statistics and continue - type Control-
Break;
                    To stop - type Control-C.
    -a              Resolve addresses to hostnames.
    -n count        Number of echo requests to send.
    -l size         Send buffer size.
    -f              Set Don't Fragment flag in packet.
    -i TTL          Time To Live.
    -v TOS          Type Of Service.
    -r count        Record route for count hops.
    -s count        Timestamp for count hops.
    -j host-list    Loose source route along host-list.
    -k host-list    Strict source route along host-list.
    -w timeout      Timeout in milliseconds to wait for each reply.
C:\>
```

Figure 8. Output of a ping Command with no Destination Address

The most useful options are highlighted in yellow. Some options do not work together, such as the **-t** and **-n** options. Other options can be used together. Experiment with the following options:

To **ping** the destination address until stopped, use the **-t** option. To stop, press <CTRL> C:

```
C:\> ping -t 192.168.254.254
Pinging 192.168.254.254 with 32 bytes of data:
Reply from 192.168.254.254: bytes=32 time<1ms TTL=63
Reply from 192.168.254.254: bytes=32 time<1ms TTL=63
Reply from 192.168.254.254: bytes=32 time<1ms TTL=63
Reply from 192.168.254.254: bytes=32 time<1ms TTL=63
Reply from 192.168.254.254: bytes=32 time<1ms TTL=63
Reply from 192.168.254.254: bytes=32 time<1ms TTL=63
Ping statistics for 192.168.254.254:
    Packets: Sent = 6, Received = 6, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
Control-C
^C
C:\>
```

Figure 9. Output of a ping Command using the -t Option

To **ping** the destination once, and record router hops, use the **-n** and **-r** options, as shown in Figure 10.
Note: Not all devices will honor the **-r** option.

```
C:\> ping -n 1 -r 9 192.168.254.254
Pinging 192.168.254.254 with 32 bytes of data:
Reply from 192.168.254.254: bytes=32 time=1ms TTL=63
    Route:      10.10.10.5 ->
           192.168.254.253 ->
           192.168.254.254 ->
                10.10.10.6 ->
            172.16.255.254
Ping statistics for 192.168.254.254:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms
C:\>
```

Figure 10. Output of a ping Command using the -n and -r Options

Task 4: Reflection

Both **ping** and **tracert** are used by network engineers to test network connectivity. For basic network connectivity, the **ping** command works best. To test latency and the network path, the **tracert** command is preferred.

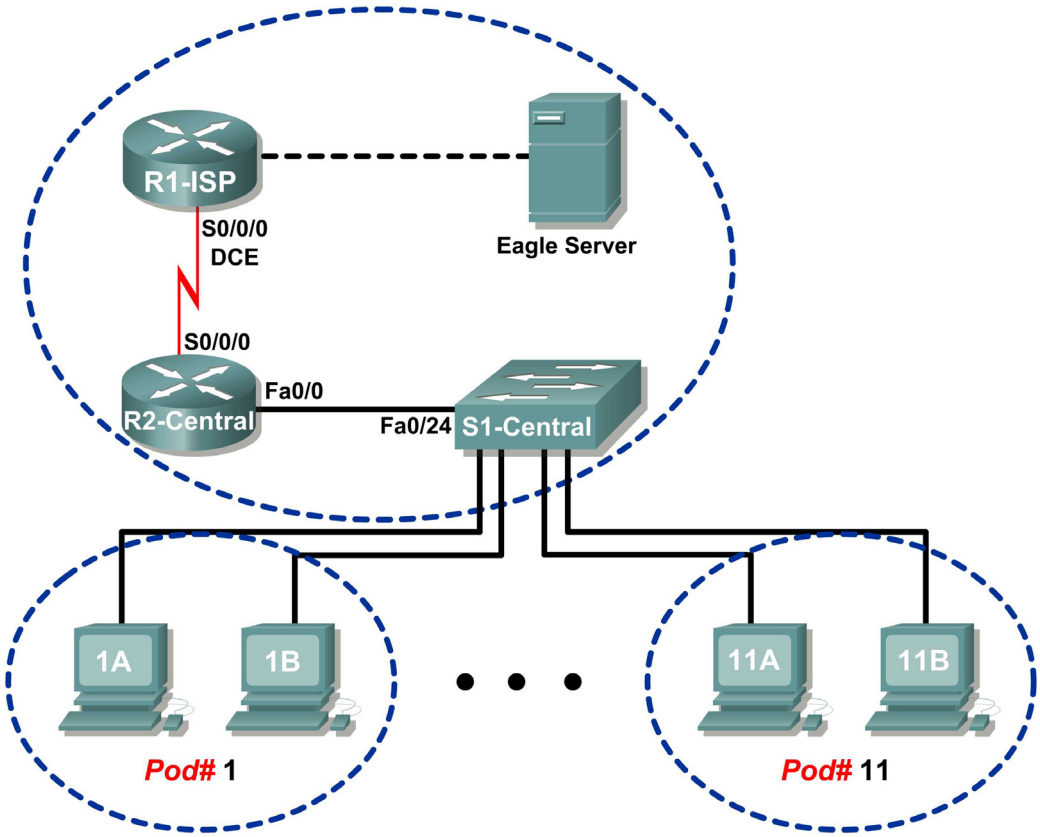
The ability to accurately and quickly diagnose network connectivity issues is a skill expected from a network engineer. Knowledge about the TCP/IP protocols and practice with troubleshooting commands will build that skill.

Task 5: Clean Up.

Unless directed otherwise by the instructor, turn off power to the host computers. Remove anything that was brought into the lab, and leave the room ready for the next class.

Lab 6.7.2: Examining ICMP Packets

Topology Diagram



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1-ISP	S0/0/0	10.10.10.6	255.255.255.252	N/A
	Fa0/0	192.168.254.253	255.255.255.0	N/A
R2-Central	S0/0/0	10.10.10.5	255.255.255.252	N/A
	Fa0/0	172.16.255.254	255.255.0.0	N/A
Eagle Server	N/A	192.168.254.254	255.255.255.0	192.168.254.253
	N/A	172.31.24.254	255.255.255.0	N/A
host*Pod#***A**	N/A	172.16.*Pod#*.1	255.255.0.0	172.16.255.254
host*Pod#***B**	N/A	172.16.*Pod#*.2	255.255.0.0	172.16.255.254
S1-Central	N/A	172.16.254.1	255.255.0.0	172.16.255.254

Learning Objectives

Upon completion of this lab, you will be able to:

- Understand the format of ICMP packets.
- Use Wireshark to capture and examine ICMP messages.

Background

The Internet Control Message Protocol (ICMP) was first defined in RFC 792, September, 1981. ICMP message types were later expanded in RFC 1700. ICMP operates at the TCP/IP Network layer and is used to exchange information between devices.

ICMP packets serve many uses in today's computer network. When a router cannot deliver a packet to a destination network or host, an informational message is returned to the source. Also, the **`ping`** and **`tracert`** commands send ICMP messages to destinations, and destinations respond with ICMP messages.

Scenario

Using the Eagle 1 Lab, Wireshark captures will be made of ICMP packets between network devices.

Task 1: Understand the Format of ICMP Packets.

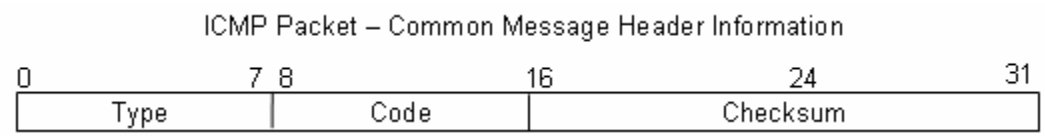


Figure 1. ICMP Message Header

Refer to Figure 1, the ICMP header fields common to all ICMP message types. Each ICMP message starts with an 8-bit Type field, an 8-bit Code field, and a computed 16-bit Checksum. The ICMP message type describes the remaining ICMP fields. The table in Figure 2 shows ICMP message types from RFC 792:

Value	Meaning
0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect
8	Echo
11	Time Exceeded
12	Parameter Problem
13	Timestamp
14	Timestamp Reply
15	Information Request
16	Information Reply

Figure 2. ICMP Message Types

Codes provide additional information to the Type field. For example, if the Type field is 3, destination unreachable, additional information about the problem is returned in the Code field. The table in Figure 3 shows message codes for an ICMP Type 3 message, destination unreachable, from RFC 1700:

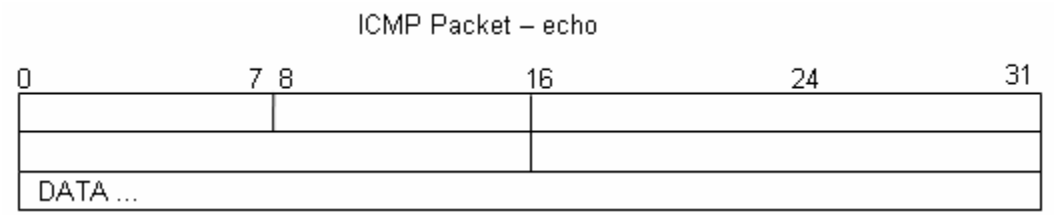
Code Value	Meaning
0	Net Unreachable
1	Host Unreachable
2	Protocol Unreachable
3	Port Unreachable
4	Fragmentation Needed and Don't Fragment was Set
5	Source Route Failed
6	Destination Network Unknown
7	Destination Host Unknown
8	Source Host Isolated
9	Communication with Destination Network is Administratively Prohibited
10	Communication with Destination Host is Administratively Prohibited
11	Destination Network Unreachable for Type of Service
12	Destination Host Unreachable for Type of Service

Figure 3. ICMP Type 3 Message Codes

Using ICMP message capture shown in Figure 4, fill in the fields for the ICMP packet echo request. Values beginning with `0x` are hexadecimal numbers:

```
Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0
  Checksum: 0x365c [correct]
  Identifier: 0x0200
  Sequence number: 0x1500
  Data (32 bytes)
```

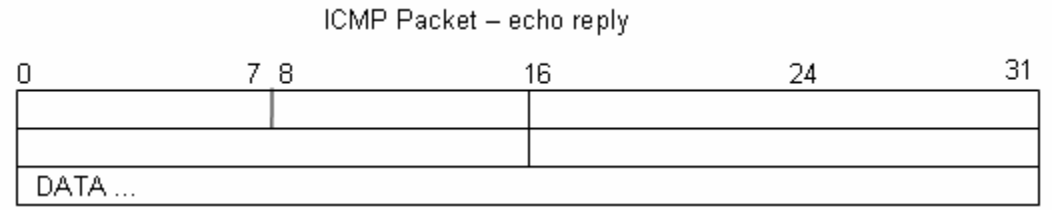
Figure 4. ICMP Packet Echo Request



Using the ICMP message capture shown in Figure 5, fill in the fields for the ICMP packet echo reply:

```
Internet Control Message Protocol
  Type: 0 (Echo (ping) reply)
  Code: 0
  Checksum: 0x3e5c [correct]
  Identifier: 0x0200
  Sequence number: 0x1500
  Data (32 bytes)
```

Figure 5. ICMP Packet Echo Reply



At the TCP/IP Network layer, communication between devices is not guaranteed. However, ICMP does provide minimal checks for a reply to match the request. From the information provided in the ICMP messages above, how does the sender know that the reply is to a specific echo?

the sequence number is used to identify the echo request.

Task 2: Use Wireshark to Capture and Examine ICMP Messages.

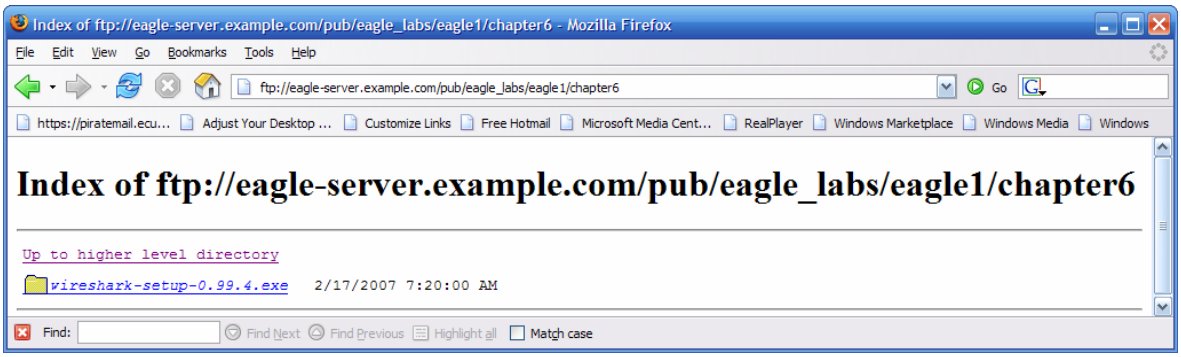


Figure 6. Wireshark Download Site

If Wireshark has not been loaded on the pod host computer, it can be downloaded from Eagle Server.

1. Open a web browser, URL FTP://eagle-server.example.com/pub/eagle_labs/eagle1/chapter6, as shown in Figure 6.
2. Right-click the Wireshark filename, click **Save Link As**, and save the file to the pod host computer.
3. When the file has been downloaded, open and install Wireshark.

Step 1: Capture and evaluate ICMP echo messages to Eagle Server.

In this step, Wireshark will be used to examine ICMP echo messages.

1. Open a Windows terminal on the pod host computer.
2. When ready, start Wireshark capture.


```
C:\> ping eagle-server.example.com
Pinging eagle-server.example.com [192.168.254.254] with 32 bytes of
data:
Reply from 192.168.254.254: bytes=32 time<1ms TTL=63
Reply from 192.168.254.254: bytes=32 time<1ms TTL=63
Reply from 192.168.254.254: bytes=32 time<1ms TTL=63
Reply from 192.168.254.254: bytes=32 time<1ms TTL=63
Ping statistics for 192.168.254.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>
```

Figure 7. Successful `ping` Replies from Eagle Server

3. From the Windows terminal, **`ping`** Eagle Server. Four successful replies should be received from Eagle Server, as shown in Figure 7.
4. Stop Wireshark capture. There should be a total of four ICMP echo requests and matching echo replies, similar to those shown in Figure 8.

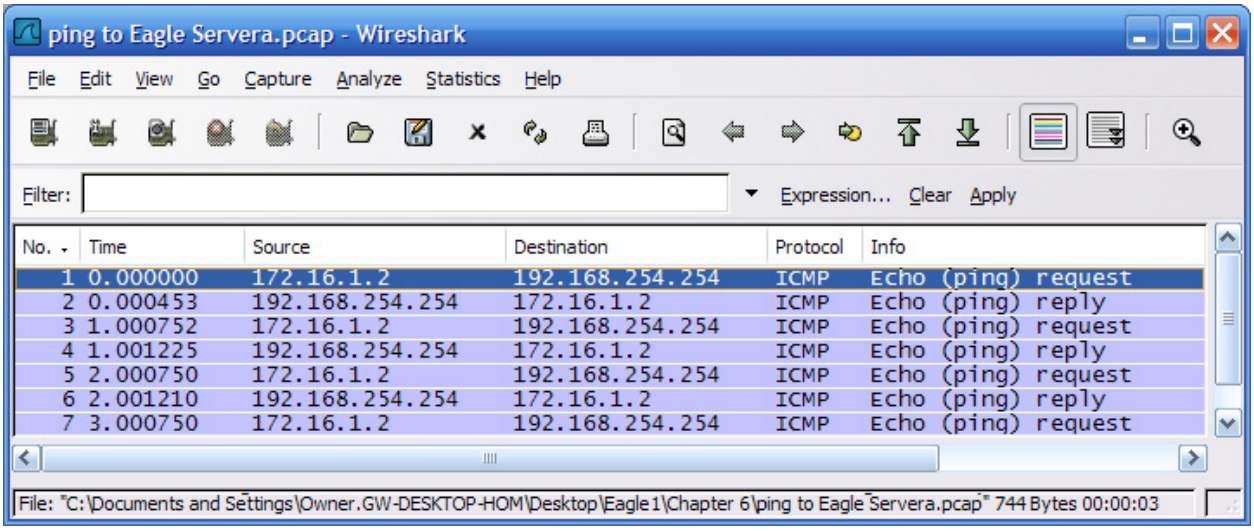


Figure 8. Wireshark Capture of `ping` Requests and Replies

Which network device responds to the ICMP echo request? 192.168.254.254

5. Expand the middle window in Wireshark, and expand the Internet Control Message Protocol record until all fields are visible. The bottom window will also be needed to examine the Data field.
6. Record information from the *first* echo request packet to Eagle Server:

Field	Value
Type	8
Code	0
Checksum	
Identifier	
Sequence number	
Data	

Are there 32 bytes of data? yes

7. Record information from the *first* echo reply packet from Eagle Server:

Field	Value
Type	0
Code	0ms
Checksum	
Identifier	
Sequence number	
Data	

Which fields, if any, changed from the echo request?

type and checksum

8. Continue to evaluate the remaining echo requests and replies. Fill in the following information from each new ping:

Packet	Checksum	Identifier	Sequence number
Request # 2			
Reply # 2			
Request # 3			
Reply # 3			
Request # 4			
Reply # 4			

Why did the Checksum values change with each new request?

sequence number

Step 2: Capture and evaluate ICMP echo messages to 192.168.253.1.

In this step, pings will be sent to a fictitious network and host. The results from the Wireshark capture will be evaluated—and may be surprising.

Try to ping IP address 192.168.253.1.

```
C:\> ping 192.168.253.1
```

```
C:\> ping 192.168.253.1
Pinging 192.168.253.1 with 32 bytes of data:
Reply from 172.16.255.254: Destination host unreachable.
Reply from 172.16.255.254: Destination host unreachable.
Reply from 172.16.255.254: Destination host unreachable.
Reply from 172.16.255.254: Destination host unreachable.
Ping statistics for 192.168.253.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>
```

Figure 9. Ping Results from a Fictitious Destination

See Figure 9. Instead of a request timeout, there is an echo response.

What network device responds to pings to a fictitious destination?

gateway router

No.	Time	Source	Destination	Protocol	Info
1	0.000000	172.16.1.2	192.168.253.1	ICMP	Echo (ping) request
2	0.000816	172.16.255.254	172.16.1.2	ICMP	Destination unreachable (Host unreachable)
3	1.000854	172.16.1.2	192.168.253.1	ICMP	Echo (ping) request
4	1.001686	172.16.255.254	172.16.1.2	ICMP	Destination unreachable (Host unreachable)
5	2.001815	172.16.1.2	192.168.253.1	ICMP	Echo (ping) request
6	2.002547	172.16.255.254	172.16.1.2	ICMP	Destination unreachable (Host unreachable)
7	3.002815	172.16.1.2	192.168.253.1	ICMP	Echo (ping) request
8	3.003588	172.16.255.254	172.16.1.2	ICMP	Destination unreachable (Host unreachable)

Figure 10. Wireshark Capture from a Fictitious Destination

Wireshark captures to a fictitious destination are shown in Figure 10. Expand the middle Wireshark window and the Internet Control Message Protocol record.

Which ICMP message type is used to return information to the sender?

3

What is the code associated with the message type?

1 host unreachable

Step 3: Capture and evaluate ICMP echo messages that exceed the TTL value.

In this step, pings will be sent with a low TTL value, simulating a destination that is unreachable. Ping Eagle Server, and set the TTL value to **1**:

```
C:\> ping -i 1 192.168.254.254
```

```
C:\> ping -i 1 192.168.254.254
Pinging 192.168.254.254 with 32 bytes of data:
Reply from 172.16.255.254: TTL expired in transit.
Reply from 172.16.255.254: TTL expired in transit.
Reply from 172.16.255.254: TTL expired in transit.
Reply from 172.16.255.254: TTL expired in transit.
Ping statistics for 192.168.254.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>
```

Figure 11. Ping Results for an Exceeded TTL

See Figure 11, which shows ping replies when the TTL value has been exceeded.

What network device responds to pings that exceed the TTL value?

gateway router

No.	Time	Source	Destination	Protocol	Info
1	0.000000	172.16.1.2	192.168.254.254	ICMP	Echo (ping) request
2	0.000701	172.16.255.254	172.16.1.2	ICMP	Time-to-live exceeded (Time to live exceeded in transit)
3	1.000003	172.16.1.2	192.168.254.254	ICMP	Echo (ping) request
4	1.000687	172.16.255.254	172.16.1.2	ICMP	Time-to-live exceeded (Time to live exceeded in transit)
5	1.999996	172.16.1.2	192.168.254.254	ICMP	Echo (ping) request
6	2.000761	172.16.255.254	172.16.1.2	ICMP	Time-to-live exceeded (Time to live exceeded in transit)
7	3.000970	172.16.1.2	192.168.254.254	ICMP	Echo (ping) request
8	3.001723	172.16.255.254	172.16.1.2	ICMP	Time-to-live exceeded (Time to live exceeded in transit)

Figure 12. Wireshark Capture of TTL Value Exceeded

Wireshark captures to a fictitious destination are shown in Figure 12. Expand the middle Wireshark window and the Internet Control Message Protocol record.

Which ICMP message type is used to return information to the sender?

type 11

What is the code associated with the message type?

code 0 time to live

Which network device is responsible for decrementing the TTL value?

routers

Task 3: Challenge

Use Wireshark to capture a **`tracert`** session to Eagle Server and then to 192.168.254.251. Examine the ICMP TTL exceeded message. This will demonstrate how the **`tracert`** command traces the network path to the destination.

Task 4: Reflection

The ICMP protocol is very useful when troubleshooting network connectivity issues. Without ICMP messages, a sender has no way to tell why a destination connection failed. Using the **`ping`** command, different ICMP message type values were captured and evaluated.

Task 5: Clean Up

Wireshark may have been loaded on the pod host computer. If the program must be removed, click **Start > Control Panel > Add or Remove Programs**, and scroll down to Wireshark. Click the filename, click **Remove**, and follow uninstall instructions.

Remove any Wireshark pcap files that were created on the pod host computer.

Unless directed otherwise by the instructor, turn off power to the host computers. Remove anything that was brought into the lab, and leave the room ready for the next class.

Activity 6.7.3: IPv4 Address Subnetting Part 1

Learning Objectives

Upon completion of this activity, you will be able to determine network information for a given IP address and network mask.

Background

This activity is designed to teach how to compute network IP address information from a given IP address.

Scenario

When given an IP address and network mask, you will be able to determine other information about the IP address such as:

- Network address
- Network broadcast address
- Total number of host bits
- Number of hosts

Task 1: For a given IP address, Determine Network Information.

Given:

Host IP Address	**172.25.114.250**
Network Mask	**255.255.0.0 (/16)**

Find:

Network Address	172.25.0.0
Network Broadcast Address	172.25.255.255
Total Number of Host Bits	16
Number of Hosts	65536

Step 1: Translate Host IP address and network mask into binary notation.

Convert the host IP address and network mask to binary:

	172	**25**	**114**	**250**
IP Address	10101100	11001000	01110010	11111010
Network Mask	11111111	11111111	00000000	00000000
	255	**255**	**0**	**0**

Step 2: Determine the network address.

1. Draw a line under the mask.
2. Perform a bit-wise AND operation on the IP address and the subnet mask.

Note: 1 AND 1 results in a 1; 0 AND anything results in a 0.

3. Express the result in dotted decimal notation.
4. The result is the network address for this host IP address, which is **172.25.0.0.**

	172	25	114	250
IP Address	10101100	11001000	01110010	11111010
Subnet Mask	11111111	11111111	00000000	00000000
Network Address	10101100	11001000	00000000	00000000
	172	**25**	**0**	**0**

Step 3: Determine the broadcast address for the network address

The network mask separates the network portion of the address from the host portion. The network address has all 0s in the host portion of the address and the broadcast address has all 1s in the host portion of the address.

	172	25	0	0
Network Add.	10101100	11001000	00000000	00000000
Mask	11111111	11111111	00000000	00000000
Broadcast.	10101100	11001000	11111111	11111111
	172	**25**	**255**	**255**

By counting the number of host bits, we can determine the total number of usable hosts for this network.

Host bits: 16

Total number of hosts:

$2^{16} = 65,536$

65,536 – 2 = 65,534 (addresses that cannot use the *all 0s* address, network address, or the *all 1s* address, broadcast address.)

Add this information to the table:

Host IP Address	**172.25.114.250**
Network Mask	**255.255.0.0 (/16)**
Network Address	172.25.0.0
Network Broadcast Address	172.25.255.255
Total Number of Host Bits Number of Hosts	16 / 65536

Task 2: Challenge

For all problems:

Create a Subnetting Worksheet to show and record all work for each problem.

Problem 1

Host IP Address	**172.30.1.33**
Network Mask	**255.255.0.0**
Network Address	172.30.0.0
Network Broadcast Address	172.30.255.255
Total Number of Host Bits	16
Number of Hosts	65536

Problem 2

Host IP Address	**172.30.1.33**
Network Mask	**255.255.255.0**
Network Address	172.30.1.0
Network Broadcast Address	172.30.1.255
Total Number of Host Bits	8
Number of Hosts	2048

Problem 3

Host IP Address	**192.168.10.234**
Network Mask	**255.255.255.0**
Network Address	192.168.10.0
Network Broadcast Address	192.168.10.255
Total Number of Host Bits	8
Number of Hosts	256

Problem 4

Host IP Address	**172.17.99.71**
Network Mask	**255.255.0.0**
Network Address	172.17.0.0
Network Broadcast Address	172.17.255.255
Total Number of Host Bits	16
Number of Hosts	65536

Problem 5

Host IP Address	**192.168.3.219**
Network Mask	**255.255.0.0**
Network Address	192.168.0.0
Network Broadcast Address	192.168.255.254
Total Number of Host Bits	0
Number of Hosts	1

Problem 6

Host IP Address	**192.168.3.219**
Network Mask	**255.255.255.224**
Network Address	192.168.3.192
Network Broadcast Address	192.168.3.223
Total Number of Host Bits	5
Number of Hosts	32

Task 3: Clean Up

Remove anything that was brought into the lab, and leave the room ready for the next class.

Activity 6.7.4: IPv4 Address Subnetting Part 2

Learning Objectives

Upon completion of this activity, you will be able to determine subnet information for a given IP address and subnetwork mask.

Background

Borrowing Bits

How many bits must be borrowed to create a certain number of subnets or a certain number of hosts per subnet?

Using this chart, it is easy to determine the number of bits that must be borrowed.

Things to remember:

- Subtract 2 for the usable number of hosts per subnet, one for the subnet address and one for the broadcast address of the subnet.

2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1,024	512	256	128	64	32	16	8	4	2	1
Number of bits borrowed:										
10	9	8	7	6	5	4	3	2	1	1
1,024	512	256	128	64	32	16	8	4	2	1
Hosts or Subnets										

Possible Subnet Mask Values

Because subnet masks must be contiguous 1's followed by contiguous 0's, the converted dotted decimal notation can contain one of a certain number of values:

Dec.	*Binary*
255	11111111
254	11111110
252	11111100
248	11111000
240	11110000
224	11100000
192	11000000
128	10000000
0	00000000

Scenario

When given an IP address, network mask, and subnetwork mask, you will be able to determine other information about the IP address such as:

- The subnet address of this subnet
- The broadcast address of this subnet
- The range of host addresses for this subnet
- The maximum number of subnets for this subnet mask
- The number of hosts for each subnet
- The number of subnet bits
- The number of this subnet

Task 1: For a Given IP Address and Subnet Mask, Determine Subnet Information.

Given:

Host IP Address	**172.25.114.250**
Network Mask	**255.255.0.0 (/16)**
Subnet Mask	**255.255.255.192 (/26)**

Find:

Number of Subnet Bits	10
Number of Subnets	1024
Number of Host Bits per Subnet	6
Number of Usable Hosts per Subnet	62
Subnet Address for this IP Address	172.25.114.192
IP Address of First Host on this Subnet	172.25.114.1
IP Address of Last Host on this Subnet	172.25.114.254
Broadcast Address for this Subnet	172.25.114.255

Step 1: Translate host IP address and subnet mask into binary notation.

	172	**25**	**114**	**250**
IP Address	10101100	11001000	01110010	11111010
	11111111	11111111	11111111	11000000
Subnet Mask	**255**	**255**	**255**	**192**

Step 2: Determine the network (or subnet) where this host address belongs.

1. Draw a line under the mask.
2. Perform a bit-wise AND operation on the IP Address and the Subnet Mask.

   Note: 1 AND 1 results in a 1' 0 AND anything results in a 0.
3. Express the result in dotted decimal notation.

4. The result is the Subnet Address of this Subnet**,** which is **172.25.114.192**

	172	**25**	**114**	**250**
IP Address	10101100	11001000	01110010	11111010
Subnet Mask	11111111	11111111	11111111	11000000
Subnet Address	10101100	11001000	01110010	11000000
	172	**25**	**114**	**192**

Add this information to the table:

Subnet Address for this IP Address	172.25.114.192

Step 3: Determine which bits in the address contain network information and which contain host information.

1. Draw the *Major Divide* (M.D.) as a wavy line where the 1s in the major network mask end (also the mask if there was no subnetting). In our example, the major network mask is 255.255.0.0, or the first 16 left-most bits.
2. Draw the *Subnet Divide* (S.D.) as a straight line where the 1s in the given subnet mask end. The network information ends where the 1s in the mask end.

			M.D.	S.D.	
IP Address	10101110	11001000	01110010	11	111010
Subnet Mask	11111111	11111111	11111111	11	000000
Subnet Add.	10001010	11001000	01110010	11	000000
			← 10 bits →		

3. The result is the Number of Subnet Bits, which can be determined by simply counting the number of bits between the M.D. and S.D., which in this case is 10 bits.

Step 4: Determine the bit ranges for subnets and hosts.

1. Label the *subnet counting range* between the M.D. and the S.D. This range contains the bits that are being incremented to create the subnet numbers or addresses.
2. Label the *host counting range* between the S.D. and the last bits at the end on the right. This range contains the bits that are being incremented to create the host numbers or addresses.

			M.D.	S.D.	
IP Address	10101110	11001000	01110010	11	111010
Subnet Mask	11111111	11111111	11111111	11	000000
Subnet Add.	10001010	11001000	01110010	11	000000
			← subnet counting range →		←host→ counting range

Step 5: Determine the range of host addresses available on this subnet and the broadcast address on this subnet.

1. Copy down all of the network/subnet bits of the network address (that is, all bits before the S.D.).
2. In the host portion (to the right of the S.D.), make the host bits all 0s except for the right-most bit (or least significant bit), which you make a 1. This gives us the *first* host IP address on this subnet, which is the *first part* of the result for *Range of Host Addresses for This Subnet*, which in the example is **172.25.114.193.**
3. Next, in the host portion (to the right of the S.D.), make the host bits all 1s except for the right-most bit (or least significant bit), which you make a 0. This gives us the *last* host IP address on this subnet, which is the last part of the result for *Range of Host Addresses for This Subnet*, which in the example is **172.25.114.254.**
4. In the host portion (to the right of the S.D.), make the host bits all 1s. This gives us the broadcast IP address on this subnet. This is the result for *Broadcast Address of This Subnet,* which in the example is **172.25.114.255.**

		M.D.	S.D.		
IP Address	10101100	11001000	01110010	11	111010
Subnet Mask	11111111	11111111	11111111	11	000000
Subnet Add.	10101100	11001000	01110010	11	000000
			← **subnet** → **counting range**		← **host** → **counting range**
First Host	10101100	11001000	01110010	11	000001
	172	25	114		193
Last Host	10101100	11001000	01110010	11	111110
	172	25	114		254
Broadcast	10101100	11001000	01110010	11	111111
	172	25	114		255

Let's add some of this information to our table:

Host IP Address	**172.25.114.250**
Major Network Mask	**255.255.0.0 (/16)**
Major (Base) Network Address	172.25.0.0
Major Network Broadcast Address	172.25.255.255
Total Number of Host Bits Number of Hosts	16 bits or 2^{16} or 65,536 total hosts 65,536 – 2 = 65,534 usable hosts
Subnet Mask	**255.255.255.192 (/26)**
Number of Subnet Bits Number of Subnets	0 /1
Number of Host Bits per Subnet Number of Usable Hosts per Subnet	6 / 62
Subnet Address for this IP Address	172.25.114.192
IP Address of First Host on this Subnet	172.25.114.193
IP Address of Last Host on this Subnet	172.25.114.254
Broadcast Address for this Subnet	172.25.114.255

Step 6: Determine the number of subnets.

The number of subnets is determined by how many bits are in the *subnet counting range* (in this example, 10 bits).

Use the formula 2^n, where *n* is the number of bits in the *subnet counting range*.

1. $2^{10} = 1024$

Number of Subnet Bits Number of Subnets (all 0s used, all 1s not used)	10 bits $2^{10} = 1024$ subnets

Step 7: Determine the number usable hosts per subnet.

The number of hosts per subnet is determined by the number of host bits (in this example, 6 bits) minus 2 (1 for the subnet address and 1 for the broadcast address of the subnet).

$2^6 – 2 = 64 – 2 = 62$ hosts per subnet

Number of Host Bits per Subnet Number of Usable Hosts per Subnet	6 bits $2^6 – 2 = 64 – 2 = 62$ hosts per subnet

Step 8: Final Answers

Host IP Address	**172.25.114.250**
Subnet Mask	**255.255.255.192 (/26)**
Number of Subnet Bits Number of Subnets	26 bits 2^{10} = 1024 subnets
Number of Host Bits per Subnet Number of Usable Hosts per Subnet	6 bits $2^6 - 2 = 64 - 2 = 62$ hosts per subnet
Subnet Address for this IP Address	172.25.114.192
IP Address of First Host on this Subnet	172.25.114.193
IP Address of Last Host on this Subnet	172.25.114.254
Broadcast Address for this Subnet	172.25.114.255

Task 2: Challenge.

For all problems:

Create a Subnetting Worksheet to show and record all work for each problem.

Problem 1

Host IP Address	**172.30.1.33**
Subnet Mask	**255.255.255.0**
Number of Subnet Bits	8
Number of Subnets	256
Number of Host Bits per Subnet	8
Number of Usable Hosts per Subnet	254
Subnet Address for this IP Address	172.30.1.0
IP Address of First Host on this Subnet	172.30.1.1
IP Address of Last Host on this Subnet	172.30.1.254
Broadcast Address for this Subnet	172.30.1.255

Problem 2

Host IP Address	**172.30.1.33**
Subnet Mask	**255.255.255.252**
Number of Subnet Bits	14
Number of Subnets	16384
Number of Host Bits per Subnet	2
Number of Usable Hosts per Subnet	2
Subnet Address for this IP Address	172.30.1.32
IP Address of First Host on this Subnet	172.30.1.33
IP Address of Last Host on this Subnet	172.30.1.34
Broadcast Address for this Subnet	172.30.1.35

Problem 3

Host IP Address	**192.192.10.234**
Subnet Mask	**255.255.255.0**
Number of Subnet Bits	0
Number of Subnets	1
Number of Host Bits per Subnet	8
Number of Usable Hosts per Subnet	254
Subnet Address for this IP Address	192.192.10.0
IP Address of First Host on this Subnet	192.192.10.1
IP Address of Last Host on this Subnet	192.192.10.254
Broadcast Address for this Subnet	192.192.10.255

Problem 4

Host IP Address	**172.17.99.71**
Subnet Mask	**255.255.0.0**
Number of Subnet Bits	0
Number of Subnets	1
Number of Host Bits per Subnet	16
Number of Usable Hosts per Subnet	65534
Subnet Address for this IP Address	172.17.0.0
IP Address of First Host on this Subnet	172.17.0.1
IP Address of Last Host on this Subnet	172.17.255.254
Broadcast Address for this Subnet	172.17.255.255

Problem 5

Host IP Address	**192.168.3.219**
Subnet Mask	**255.255.255.0**
Number of Subnet Bits	0
Number of Subnets	1
Number of Host Bits per Subnet	8
Number of Usable Hosts per Subnet	254
Subnet Address for this IP Address	192.168.3.0
IP Address of First Host on this Subnet	192.168.3.1
IP Address of Last Host on this Subnet	192.168.3.254
Broadcast Address for this Subnet	192.168.3.255

Problem 6

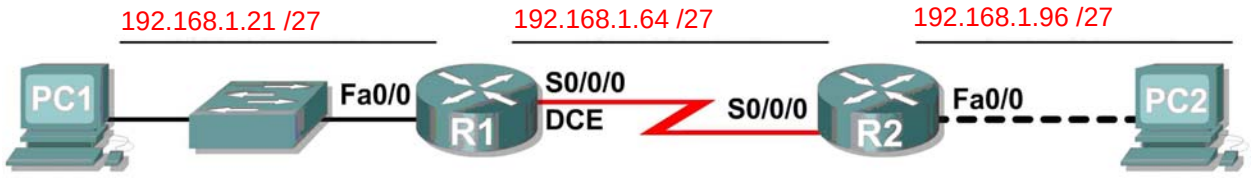
Host IP Address	**192.168.3.219**
Subnet Mask	**255.255.255.252**
Number of Subnet Bits	6
Number of Subnets	64
Number of Host Bits per Subnet	2
Number of Usable Hosts per Subnet	2
Subnet Address for this IP Address	192.168.3.216
IP Address of First Host on this Subnet	192.168.3.217
IP Address of Last Host on this Subnet	192.168.3.218
Broadcast Address for this Subnet	192.168.3.219

Task 3: Clean Up

Remove anything that was brought into the lab, and leave the room ready for the next class.

Lab 6.7.5: Subnet and Router Configuration

Topology Diagram



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	Fa0/0	192.168.1.33	255.255.255.224	N/A
	S0/0/0	192.168.1.65	255.255.255.224	N/A
R2	Fa0/0	192.168.1.97	255.255.255.224	N/A
	S0/0/0	192.168.1.94	255.255.255.224	N/A
PC1	NIC	192.168.1.62	255.255.255.224	192.168.1.33
PC2	NIC	192.168.1.126	255.255.255.224	192.168.1.97

Learning Objectives

Upon completion of this lab, you will be able to:

- Subnet an address space per given requirements.
- Assign appropriate addresses to interfaces and document.
- Configure and activate Serial and FastEthernet interfaces.
- Test and verify configurations.
- Reflect upon and document the network implementation.

Scenario

In this lab activity, you will design and apply an IP addressing scheme for the topology shown in the Topology Diagram. You will be given one address block that you must subnet to provide a logical addressing scheme for the network. The routers will then be ready for interface address configuration according to your IP addressing scheme. When the configuration is complete, verify that the network is working properly.

Task 1: Subnet the Address Space.

Step 1: Examine the network requirements.

You have been given the 192.168.1.0/24 address space to use in your network design. The network consists of the following segments:

- The LAN connected to router R1 will require enough IP addresses to support 15 hosts.
- The LAN connected to router R2 will require enough IP addresses to support 30 hosts.
- The link between router R1 and router R2 will require IP addresses at each end of the link.

The plan should have equal size subnets and use the smallest subnet sizes that will accommodate the appropriate number of hosts.

Step 2: Consider the following questions when creating your network design.

How many subnets are needed for this network? 3

What is the subnet mask for this network in dotted decimal format? 255.255.255.224

What is the subnet mask for the network in slash format? /27

How many usable hosts are there per subnet? 30

Step 3: Assign subnetwork addresses to the Topology Diagram.

1. Assign second subnet to the network attached to R1.
2. Assign third subnet to the link between R1 and R2.
3. Assign fourth subnet to the network attached to R2.

Task 2: Determine Interface Addresses.

Step 1: Assign appropriate addresses to the device interfaces.

1. Assign the first valid host address in second subnet to the LAN interface on R1.
2. Assign the last valid host address in second subnet to PC1.
3. Assign the first valid host address in third subnet to the WAN interface on R1.
4. Assign the last valid host address in third subnet to the WAN interface on R2.
5. Assign the first valid host address in fourth subnet to the LAN interface of R2.
6. Assign the last valid host address in fourth subnet to PC2.

Step 2: Document the addresses to be used in the table provided under the Topology Diagram.

Task 3: Configure the Serial and FastEthernet Addresses.

Step 1: Configure the router interfaces.

Configure the interfaces on the R1 and R2 routers with the IP addresses from your network design. Please note, to complete the activity in Packet Tracer you will be using the Config Tab. When you have finished, be sure to save the running configuration to the NVRAM of the router.

Step 2: Configure the PC interfaces.

Configure the Ethernet interfaces of PC1 and PC2 with the IP addresses and default gateways from your network design.

Task 4: Verify the Configurations.

Answer the following questions to verify that the network is operating as expected.

From the host attached to R1, is it possible to ping the default gateway? yes

From the host attached to R2, is it possible to ping the default gateway? yes

From the router R1, is it possible to ping the Serial 0/0/0 interface of R2? yes

From the router R2, is it possible to ping the Serial 0/0/0 interface of R1? yes

Task 5: Reflection

Are there any devices on the network that cannot ping each other?

yes

What is missing from the network that is preventing communication between these devices?

switch

