# The OSI Model

## OBJECTIVE:

### CompTIA Network + Domain:

Domain 1.0: Networking Concepts

### CompTIA Network + Objective:

Objective 1.1: Explain devices, applications, protocols, and services at their appropriate OSI layers.

## OVERVIEW:

This lab will utilize Wireshark to review network traffic. Wireshark is a network protocol analyzer licensed under GNU General Public License. A network protocol analyzer is used to capture data packets on a network. Students will review several layers of the OSI model during this lab. Students will be able to describe the encapsulation process and the function of specific protocols that operate within particular layers of the OSI model.

## OUTCOMES:

In this lab you will learn to:

1. Explain the application, presentation, and session layers.
2. Explain the transport layer.
3. Explain the network layer.
4. Explain the data link layer.
5. Explain the physical layer.

| Key Term | Description |
|---|---|
| Connection-oriented data transfer | a transfer of data that requires the establishment of a connection between communicating endpoints, before the transfer can begin |
| Connectionless data transfer | a transfer of data that is serviced without requiring a verified session and without guaranteeing delivery of data |
| De-encapsulation | the process of each layer of the OSI model removing the control information headers on incoming information for the corresponding layer at the destination |
| Encapsulation | the process of each layer of the OSI model adding control information headers to outgoing network data |
| IANA | Internet Assigned Numbers Authority; a government-funded group responsible for managing IP address allocation and the Domain Name System (DNS) |
| IEEE | Institute of Electrical and Electronics Engineers; one of the leading standards-making organizations in the world |

| Key Term | Description |
|---|---|
| IP | Internet Protocol; a core protocol of the TCP/IP suite that resides at the Network layer of the OSI model and provides information about how packets should be routed between networks |
| MAC address | Media Access Control; the physical address burned into the ROM of an Ethernet network card; used by switches at the Data Link layer of the OSI model to move information between nodes on the same network |
| OSI | Open System Interconnect; developed by the International Standards Organization (ISO) |
| OUI | Organizationally Unique Identifier; the first 24 bits (or 3 bytes) of a MAC address assigned by IEEE that identifies the network card's manufacturer |
| PDU | Protocol Data Unit; a term used to describe the product of encapsulation at a given layer of the OSI model |
| TCP | Transmission Control Protocol; the connection-oriented protocol of the TCP/IP suite that resides at the Transport layer of the OSI model |
| UDP | User Datagram Protocol; the connectionless protocol of the TCP/IP suite that resides at the Transport layer of the OSI model |
| Wireshark | a network protocol analyzer. It lets you capture and interactively browse the traffic running on a computer network. It has a rich and powerful feature set and is world's most popular tool of its kind. It runs on most computing platforms including Windows, OS X, Linux, and UNIX. Network professionals, security experts, developers, and educators around the world use it regularly. It is freely available as open source, and is released under the GNU General Public License version 2." Reference: http://www.wireshark.org |

# Reading Assignment

## Introduction

In this lab, you will be using Wireshark to explore the layers of the Open Systems Interconnect, or OSI, Model. Figure 1 shows the lab topology for this lab. You will be using a pcap file with previously captured network activity in Wireshark to explore that network traffic and illustrate each of the seven layers of the OSI Model.


**FIGURE 1 – LAB TOPOLOGY**

## OSI Model

The Open System Interconnection (OSI) Model is a vendor neutral conceptual model that consists of seven layers from the physical layer to the application layer. Figure 2 shows the seven layers of the OSI model.

The data at each of the layers of the OSI model is called the protocol data unit (PDU) which includes the message along with a header. The application, presentation, and session layers' PDU is the data. The transport layer's PDU is called either the segment or datagram. The network layer's PDU is called the packet. The data link layer's PDU is called the frame, and the physical layer's PDU is the bits. As you move up the layers, encapsulation occurs where each layer add a header to its role in the communication process. When the receiver receives the complete PDU, decapsulation occurs where the header informs the functions of each layer.

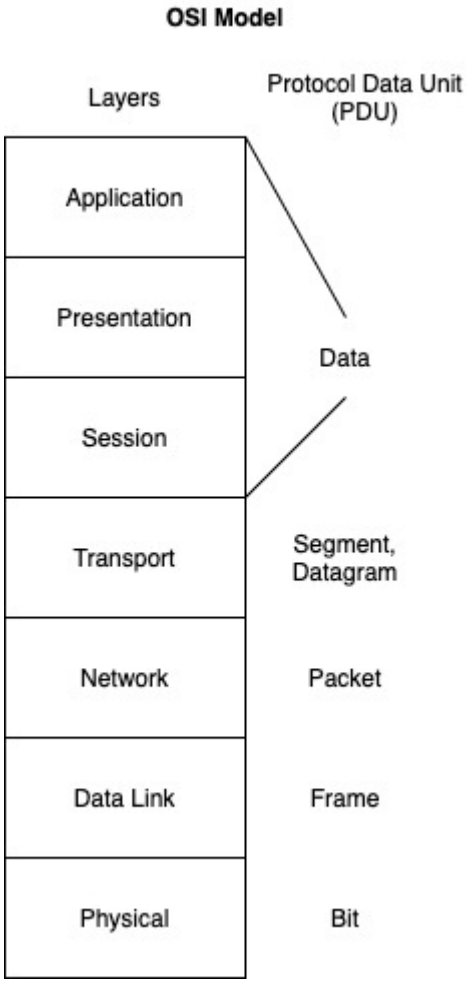



**FIGURE 2 – OSI MODEL**

## TCP/IP

Compared to the OSI Model, the Transmission Control Protocol/Internet Protocol (TCP/IP) networking model consists of four layers: application, transport, network, and data link. Figure 3 shows the different TCP/IP layers compared to the OSI model. It is important to note that the OSI Model is conceptual, whereas the TCP/IP model is the actual implementation of how the data flows. The TCP/IP model combines the functions of the first three OSI Model layers and also combines the last two layers (data link and physical). The names of the PDUs at each layer still apply.  Services run at the application layer and interact with the transport layer using ports. Port numbers are assigned to different services on the operation system. Services, such as File Transfer Protocol (FTP), telnet, Hypertext Transport Protocol (HTTP), and others use unique port numbers assigned to them by the operating system. FTP has a port number of 21, telnet uses the port number of 23, and HTTP has a port number of 80. These port numbers are how TCP/IP knows how to communicate from the transport layer to the application layer. TCP/IP was not initially designed with security in mind so some applications are configured by default to send traffic over the network in plaintext. There are relatively newer services and more widely used protocols that use encryption like

Secure Shell (SSH) and Hypertext Transport Protocol Secure (HTTPS) that are used in place of these older, less secure protocols.

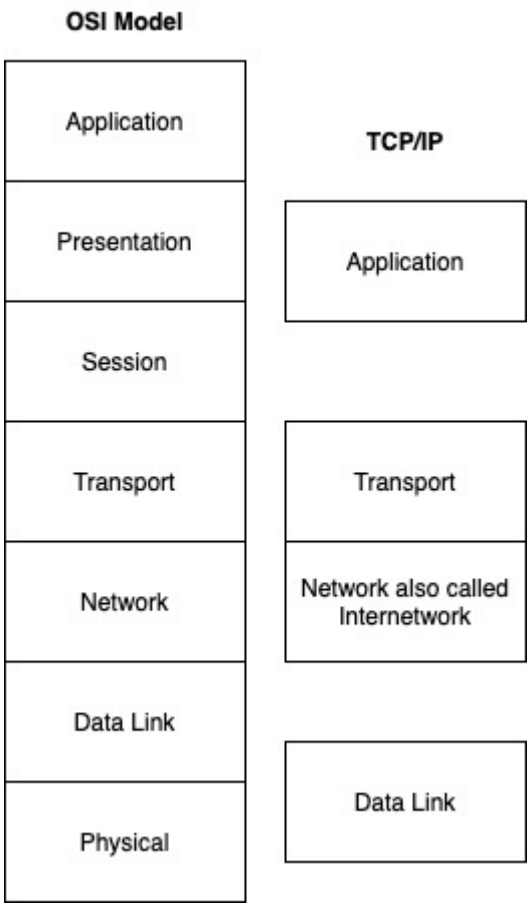



**FIGURE 3 – TCP/IP NETWORKING MODEL VERSUS THE OSI MODEL**

There are several protocols used in this lab which will have an image of the header format to assist in analyzing network traffic when you are using Wireshark.

## Application, Presentation, and Session Layers

In the OSI Model, the application layer is where the app is and the application on the system resides. The presentation layer translates the application data into the network representation that is recognizable by the communication system. The session layer initiates and manages the communication on the network.

## Transport

In the OSI Model, the PDU is called either a segment or datagram. The transport layer manages the reliable transport of segments on the network including segmentation, acknowledgment, and multiplexing. There are two protocols at this layer in TCP/IP: transmission control protocol (TCP) and user datagram protocol (UDP).

## Transport Control Protocol (TCP)

TCP is a protocol that sits at the transport layer of the TCP/IP stack. It is a reliable, ordered, connection oriented, and error checked. The job of the TCP protocol is to make sure that a connection is created between the source and destination host and reliably sends packets over the network. TCP works in three phases: connection setup, data transmission, and connection termination. Port numbers are assigned to application layer protocols to allow for applications to talk to each other from source to destination. Figure 4 shows the TCP protocol.

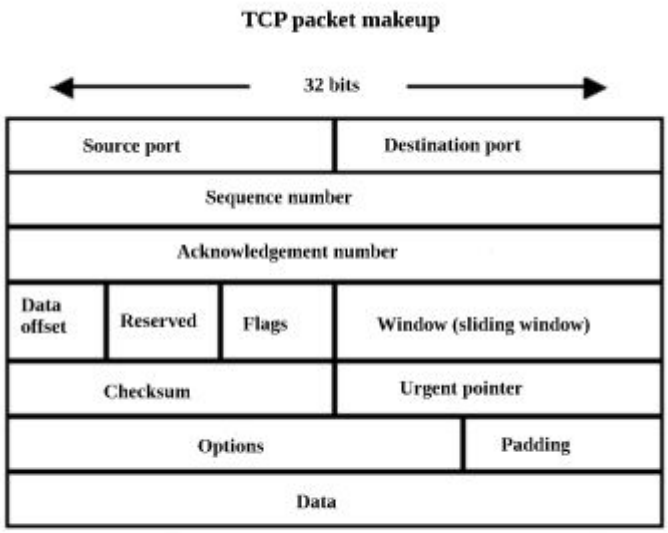



FIGURE 4 – TCP PROTOCOL (SOURCE: TCP)

## User Datagram Protocol (UDP)

UDP is a protocol that sits at the transport layer of the TCP/IP stack. It is a connectionless protocol. It provides minimal error checking unlike TCP. It also allows for port numbers to communicate with application layer protocols. Figure 5 shows the UDP protocol.

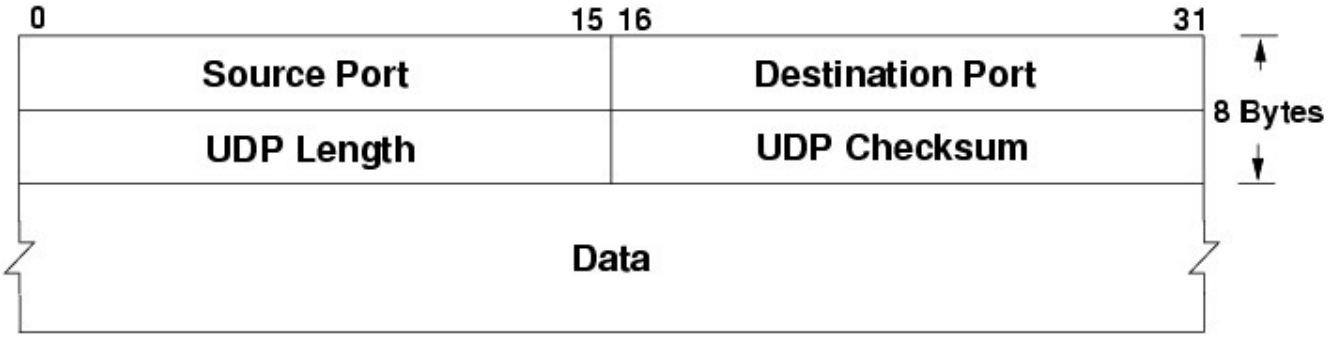



FIGURE 5 – UDP PROTOCOL (SOURCE: UDP)

## Network

In the OSI Model, the network layer is responsible for addressing, routing, and traffic control. In TCP/IP, the Internet Protocol (IP) handles this function.

## Internet Protocol (IP)

IP version 4 is a connectionless network layer protocol that transmits packets from a source host to a destination host. It uses a 32-bit address space and usually represented in decimal dotted notation (e.g., 192.153.10.1). One of the functions of the IP is a routing function that allows for communications between hosts on a local area network (LAN) and a Wide Area Network (WAN). The successor to IPv4 is IPv6. Figure 6 shows the IPv4 protocol.

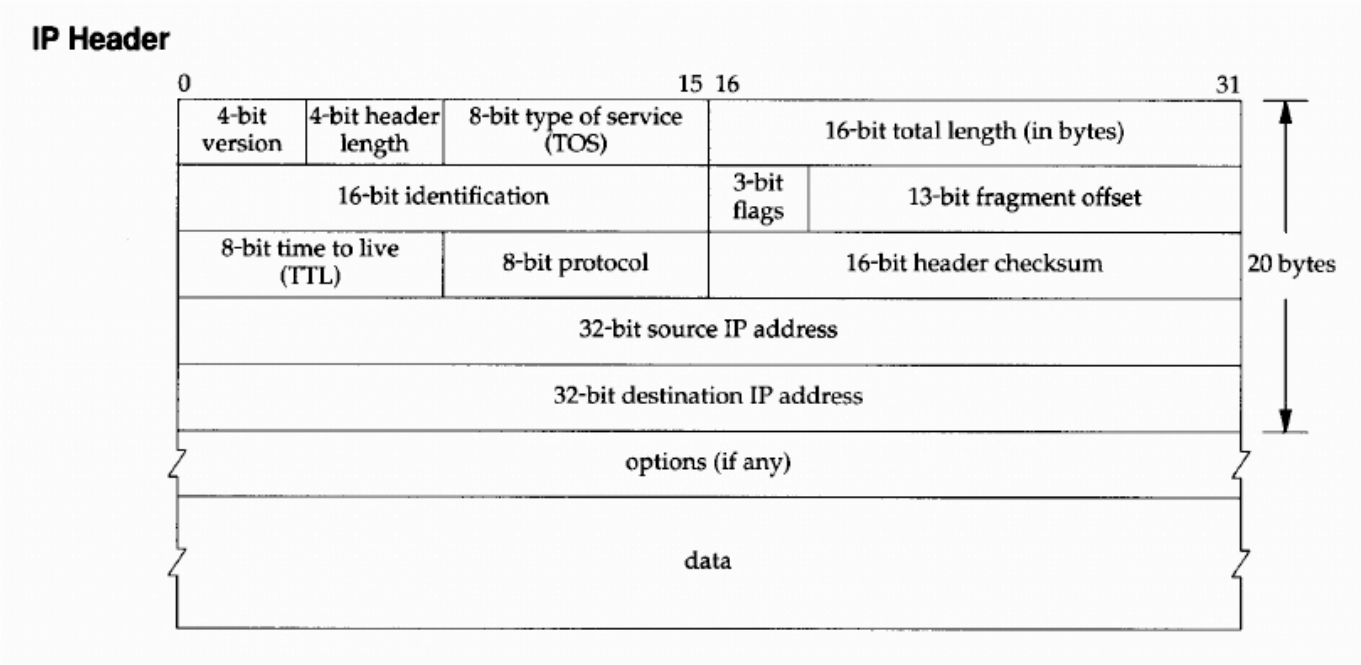



FIGURE 6 – IP PROTOCOL (SOURCE: IP)

## Internet Protocol (IPv6)

IPv6 is the successor to IP which is also known as IPv4. It is an upgrade to IPv4 to allow for more addressing, and for addresses, it uses a hexadecimal address. IPv6 also allows for a much larger 128-bit address space. Notice the difference in the size of the IP address. Figure 7 shows IPv6 protocol.


FIGURE 7 – IPV6 (SOURCE: IPV6)

## Data Link and Physical

In the OSI Model, the data link layer reliable transmission of data frames between two nodes on a network. The physical layer is responsible for the raw bit streams over a physical medium.

## Ethernet

Ethernet is an IEEE 802.3 standard used for networks. It handles the communication over wired switched network today. The frame format is shown in Figure 8. The preamble is there to help devices synchronize bit patterns. You have the source Medium Access Control (MAC) addresses which are unique to each network interface card (NIC). The NIC is the interface between the operating system and the physical

network. The last part of the frame is the Cyclical Redundancy Check (CRC) which handles error detection of the data being transmitted.

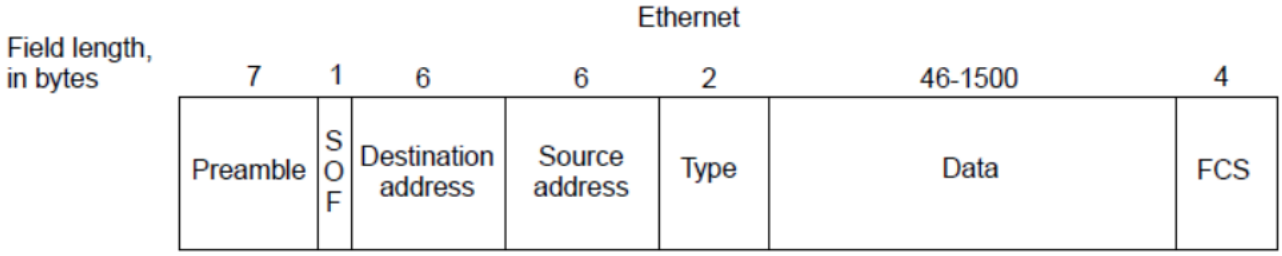



FIGURE 8 – ETHERNET FRAME (SOURCE: ETHERNET)

## Examining Protocol Traffic in Wireshark

Wireshark is a network protocol analyzer. It allows you to inspect and capture packets on your network. It allows you to inspect the traffic that is transmitting on your network.

The format for a packet that is transmitted over a network usually looks like in Figure 9.

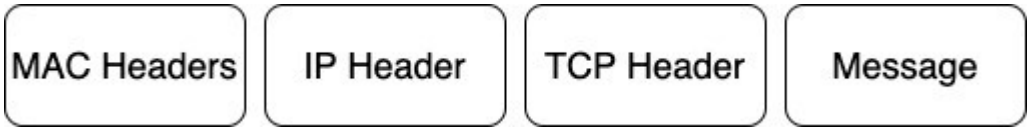



FIGURE 9 – PACKET FORMAT

This relates to the layers in the TCP/IP protocol stack. Media Access Control (MAC) header is Ethernet, Internet Protocol (IP) header is the network/Internet layer, TCP header is the transport layer, and the message is the application layer. When a message is transmitted over the network, it encapsulates the header from each of the layers before it transmits onto the network. When the message is received, the headers are stripped off as it works its way up the protocol stack to the application. Figure 10 illustrates how a message flows from the client to the server.

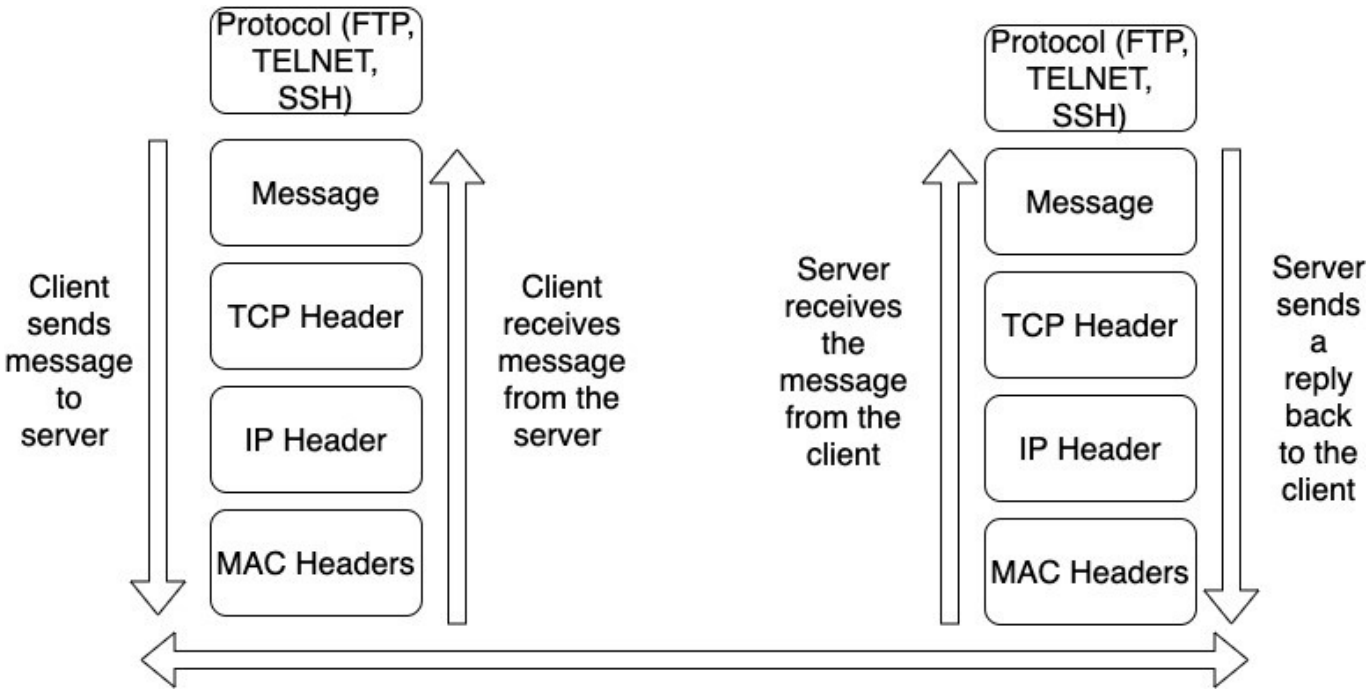



FIGURE 10 – MESSAGE FLOW FROM CLIENT TO SERVER AND BACK

Wireshark provides a user interface that allows you to filter your network traffic and analyze that traffic. A system administrator can use Wireshark if he or she suspects there might be nefarious traffic that the firewall and intrusion detection system is not detecting. A system administrator needs to know protocols in depth to grasp the information being transmitted on the network. Figure 11 shows the user interface for Wireshark. You open a network capture file, and the first step is to filter the traffic using a DisplayFilter.

A DisplayFilter allows you to only see traffic that you want to see. You can filter on items like the TCP port number, the protocol type, IP addresses, etc. For more information on DisplayFilters, see this link. To fully appreciate the details of the headers of the different protocols at the different layers, you need to review the header information. Wikipedia is a good source of header information for the different protocols used on a network. Once the filter is set, the results appear in #2. As you change the DisplayFilter, you can filter the display to show only the relevant related traffic. When you click on a packet, the packet info appears in #3. Details about the selected link are provided in the second part of the window. You can examine the details of that particular part of the captured data. #4 of the screenshot shows the file in hexadecimal format on the left side of the pane.

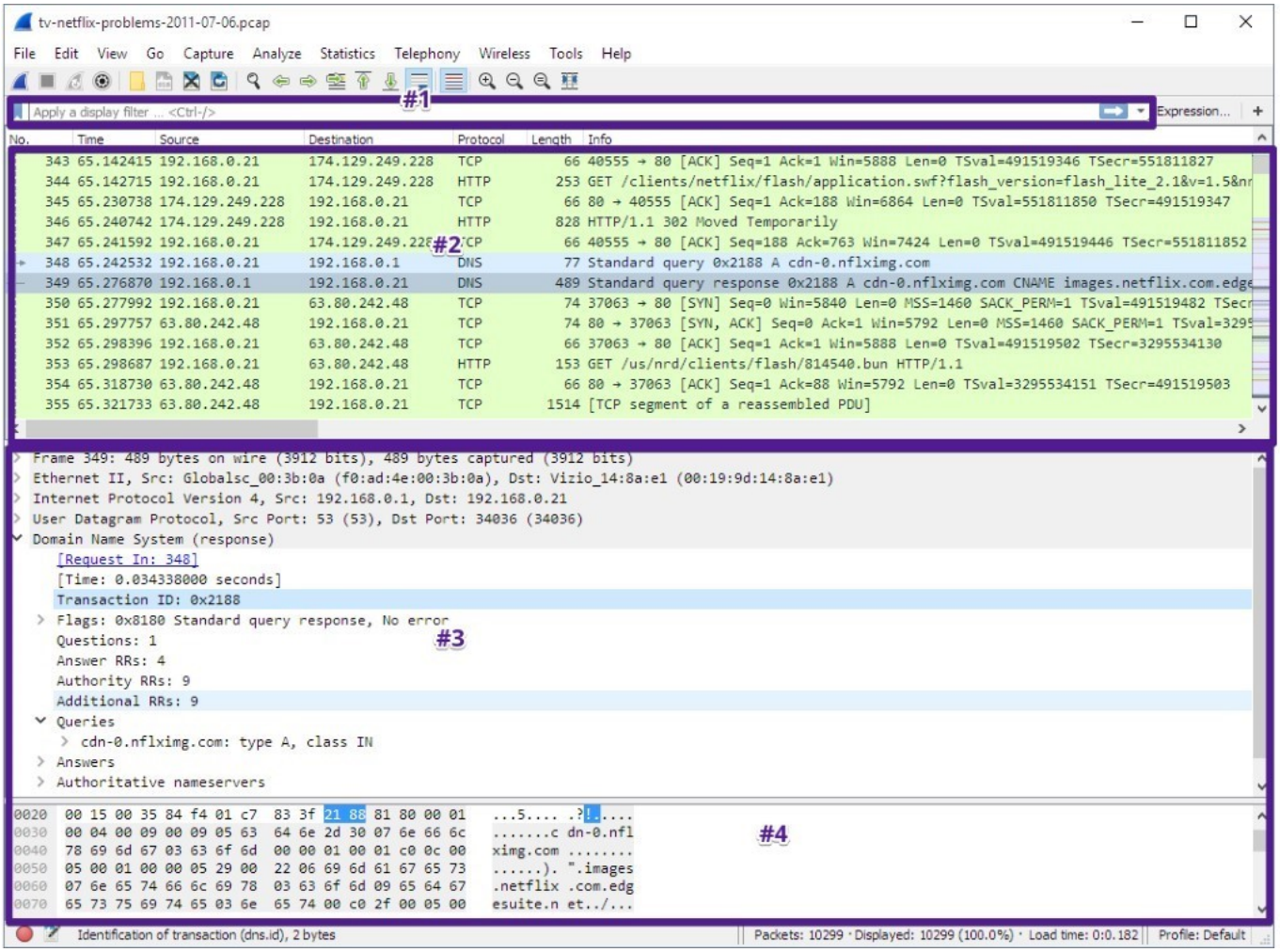


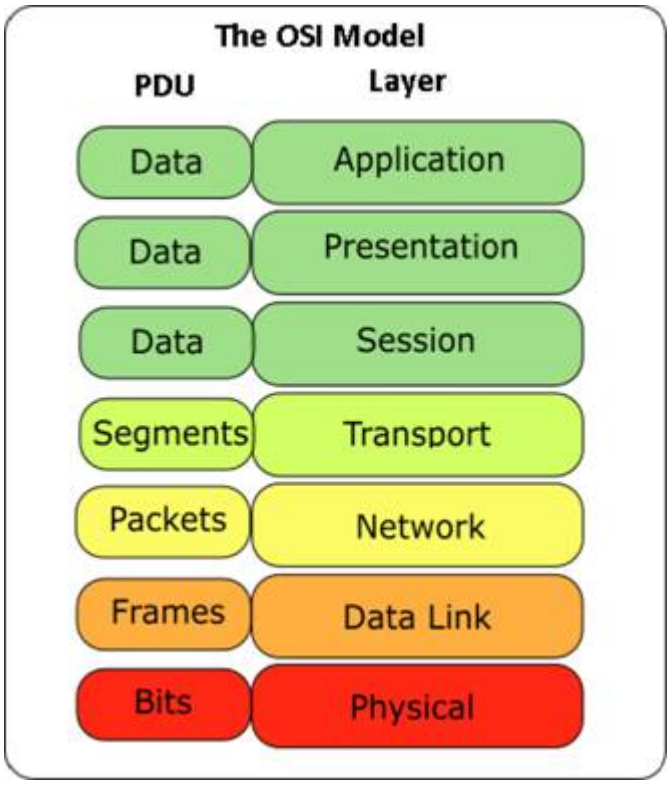

**FIGURE 11 – WIRESHARK INTERFACE**

# CONCLUSION:

In this lab, you will be using Wireshark to analyze different protocols at different layers of the OSI and TCP/IP models.

# Review of the OSI Model and Wireshark

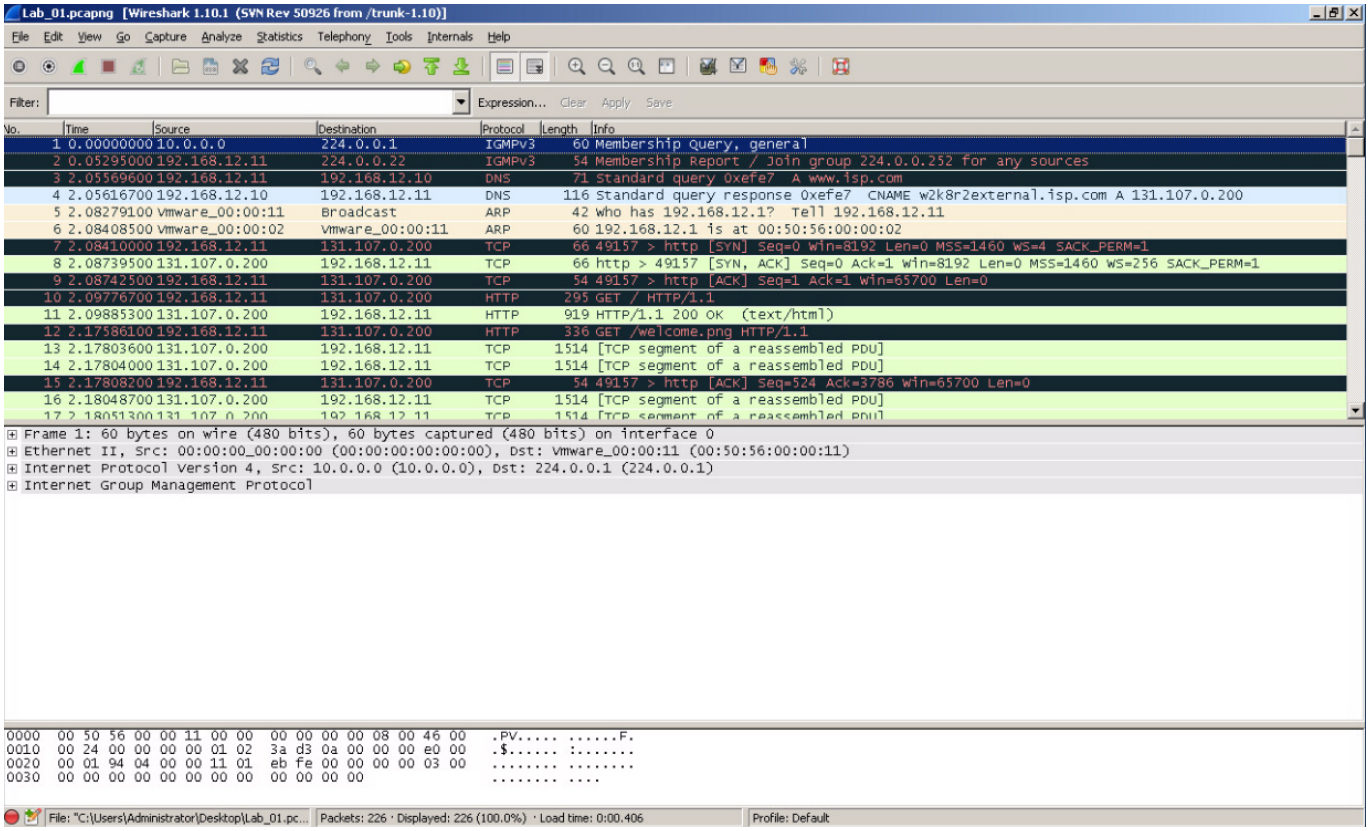
The Open System Interconnection, or OSI, model defines a framework through which networking protocols (or protocol suites) can be implemented. The OSI model consists of seven layers. Each layer has its own responsibility within the communication process. Hosts that have data to send over the network pass the data through each of the seven layers, starting at the top, until the last layer is reached. Each layer adds the information it needs to the data in a process known as encapsulation. The information added at each layer usually comes in the form of a header specific to the protocol in use at that layer. As the data is manipulated at each layer, a new name is given to it, as to associate it with the specific layer. These new data pieces are called Protocol Data Units (PDU). The seven layers of the OSI model and the PDU associated with the layer is shown.




THE OSI MODEL

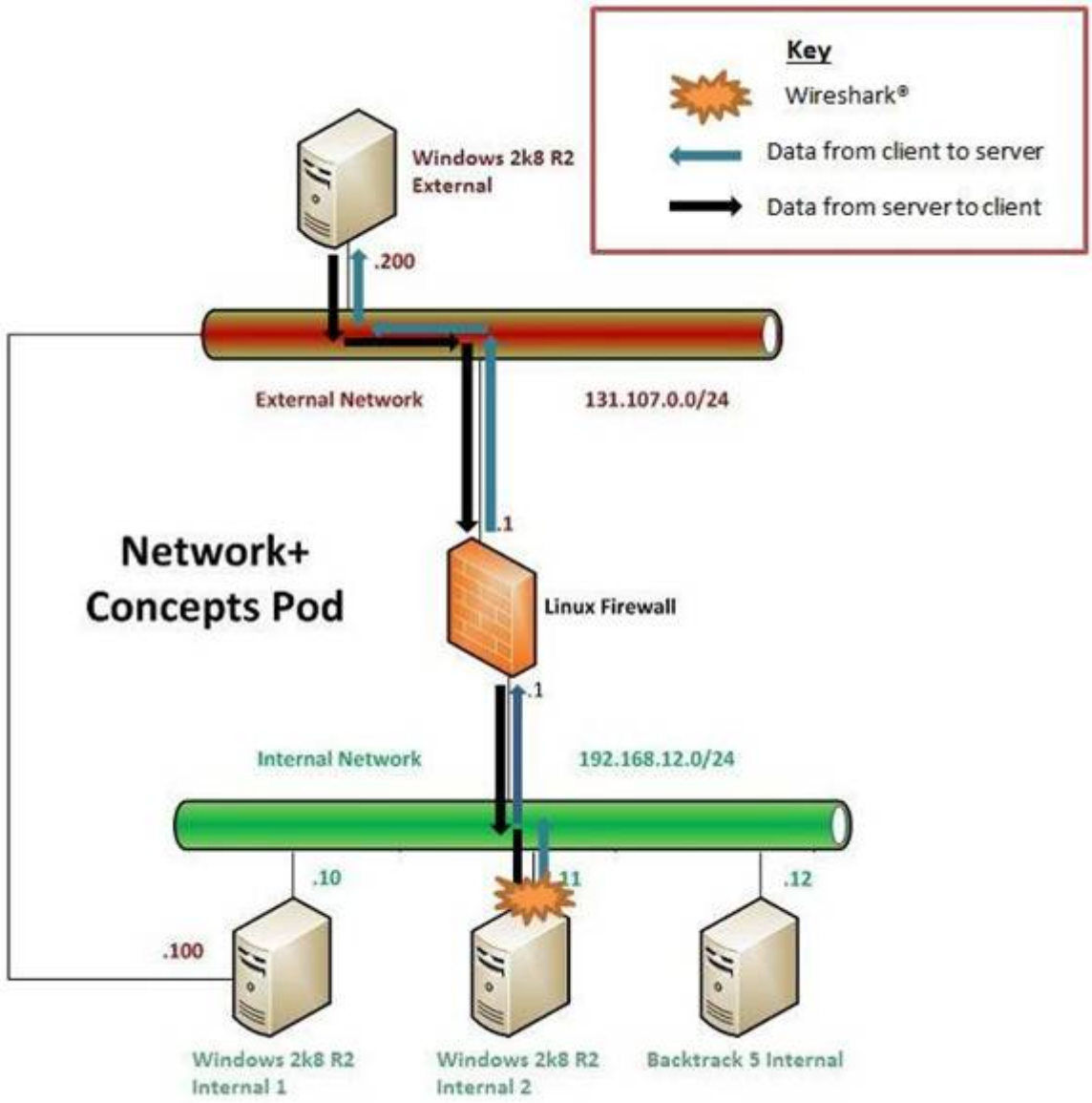
Once the data has reached the physical layer of the OSI model, it is transmitted onto the networking media and sent to the destination host. The destination host passes the data back up through the layers of the OSI model with each layer processing and removing its header. This process is known as de-encapsulation. This process continues up the layers of the OSI model until the receiving host's application processes the data.

Wireshark is a network protocol analyzer that allows you to capture and interactively browse the traffic running on a computer network. With Wireshark, users can view the encapsulation and de-encapsulation process for any captured network conversation. Wireshark runs interactively on one of the client computers and works by processing every data packet it receives on its network interface even if that packet is not destined for the client system running Wireshark. A network interface functioning in this manner is said to be operating in promiscuous mode. It does not interfere with normal network communication. Instead, it simply displays all received data in the program's capture window.




### WIRESHARK

From this window, a user can view the contents of any captured packet to reveal the details of a network conversation. In this lab, you will view a network conversation between a web client application and a web server for the request of a webpage at the URL http://www.isp.com. The host running the web client application, requesting the webpage is the Internal 192.168.12.11 Windows Server machine in the pod topology. This is also the machine running the Wireshark utility. The web server responding to the request for the webpage is the Windows 2k8 R2 External machine in the pod topology.




**POD TOPOLOGY**

This lab serves as a demonstration of Wireshark's ability to capture and view this process. Each layer of the OSI model will be identified, and the data associated with that layer will be viewed in its raw format. It is not expected that you will become a network expert at the conclusion of this lab; instead, this lab serves to give you an understanding of how the OSI model functions and to demonstrate the powerful capabilities of the Wireshark utility.

# DISCUSSION QUESTIONS:

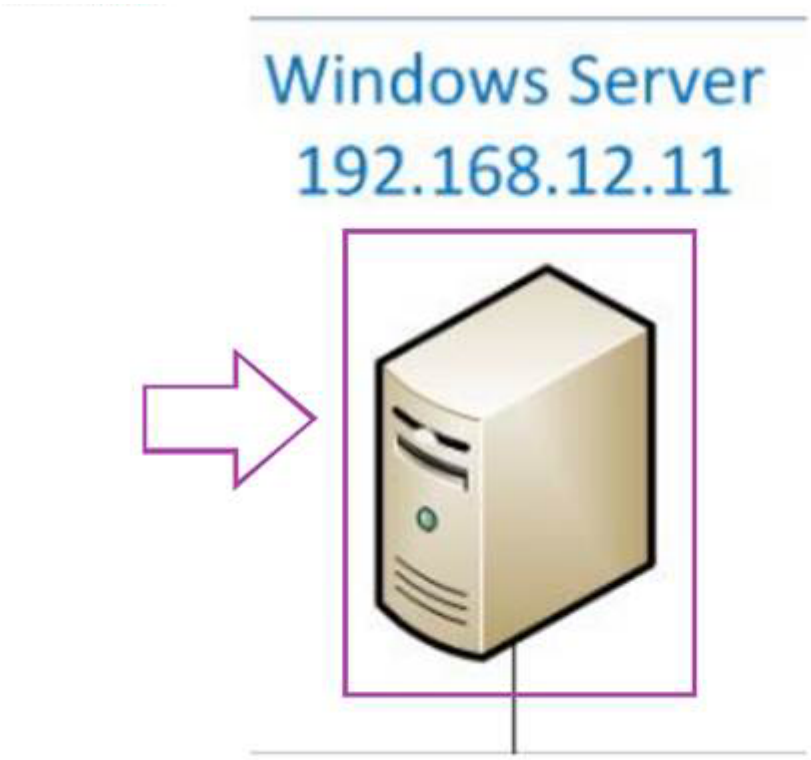
1. What is the OSI model?
2. What is Wireshark?
3. What is promiscuous mode?

# Reviewing the Application, Presentation, and Session Layers

Many protocols operate at the application, presentation, and session layers of the OSI model. The top three layers of the OSI are often looked at from the perspective of the TCP/IP model which encompasses all three layers into one layer labeled application. These three layers operate on the data that is being formed and readied to be packaged. The PDU associated with information created by any of the top three layers of the OSI model is referred to as data. The protocols at these layers prepare the data by formatting it based on the network service or application being used, encrypting and encoding the data, and controlling the dialog between the end system applications. Examples of network services, protocols, and client requests interfacing at these layers include File Transfer Protocol (FTP), Telnet, and Hypertext Transfer Protocol (HTTP).

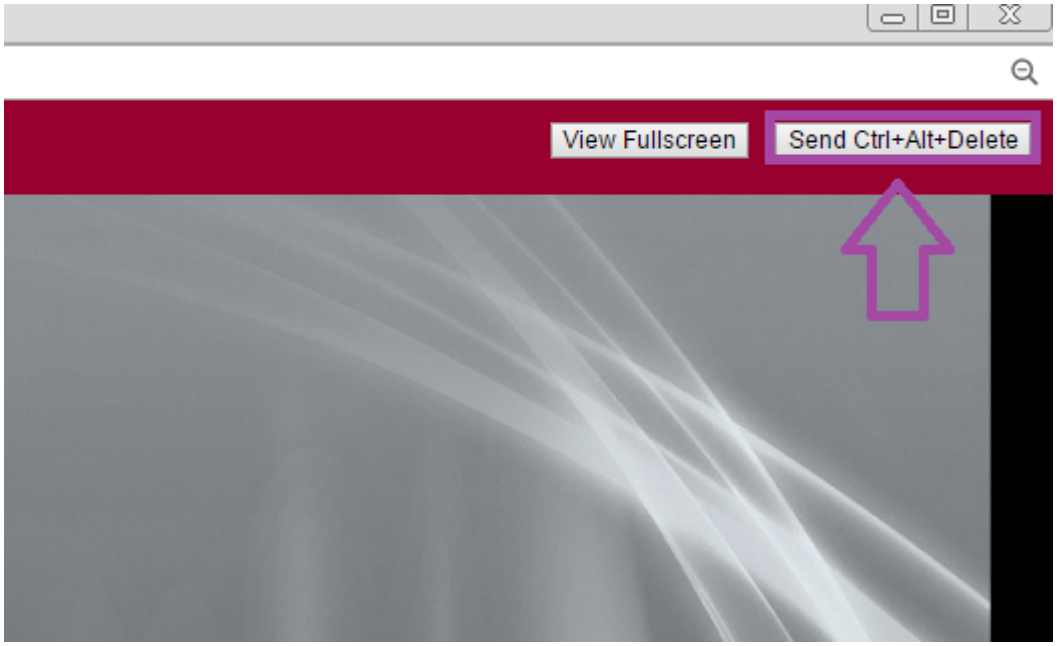
## Data Link Protocol Data Unit

1. **Click** on the Internal 192.168.12.11 Windows Server icon in the topology diagram.




**INTERNAL 192.168.12.11 WINDOWS SERVER**

2. After the machine boots up, **click** the Send Ctrl+Alt+Delete button on the upper-right corner.

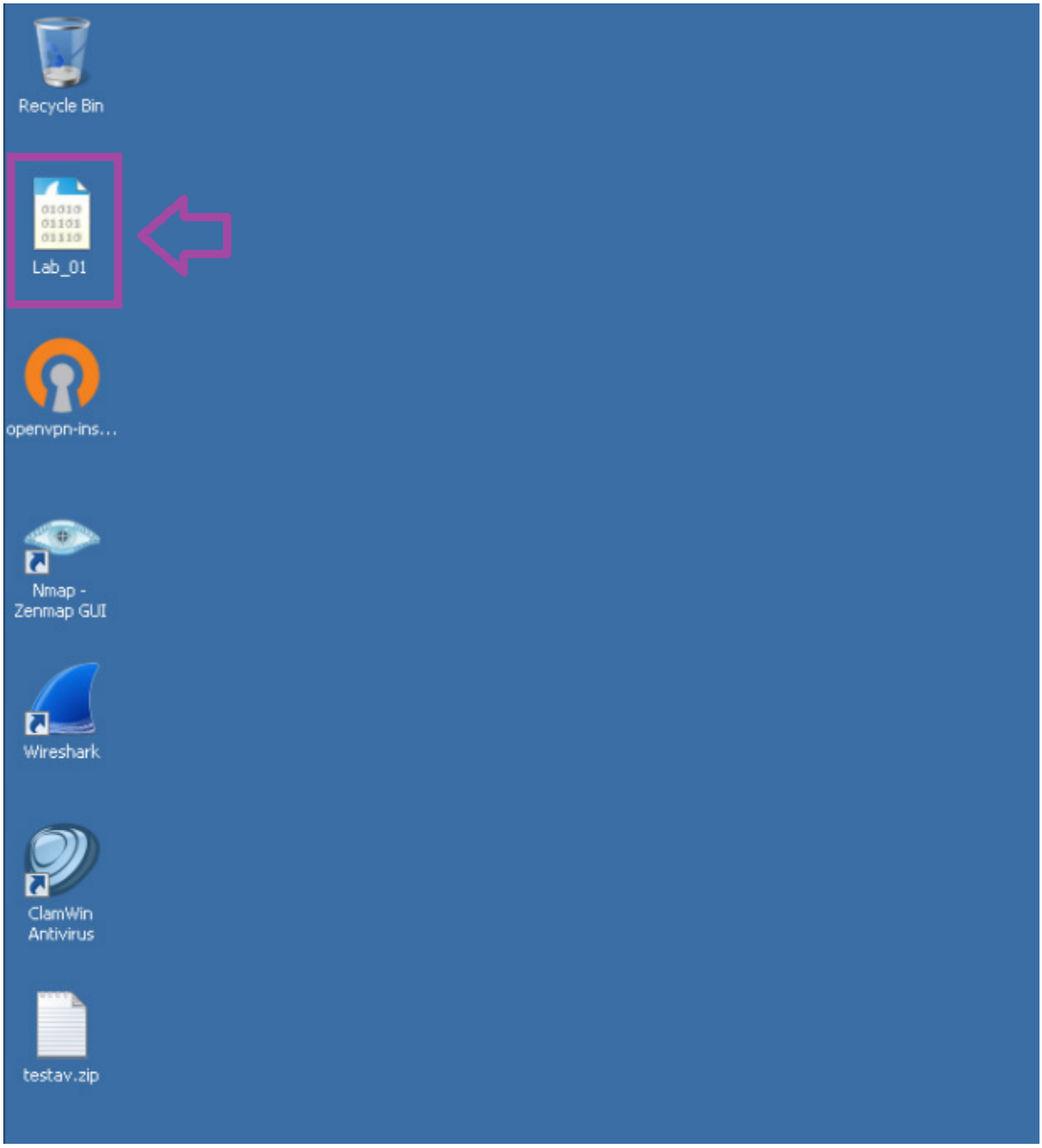



SEND CTRL+ALT+DELETE BUTTON

3. In the password text box, **type** `P@ssw0rd` and **press** Enter to log into the Internal 192.168.12.11 Windows Server.

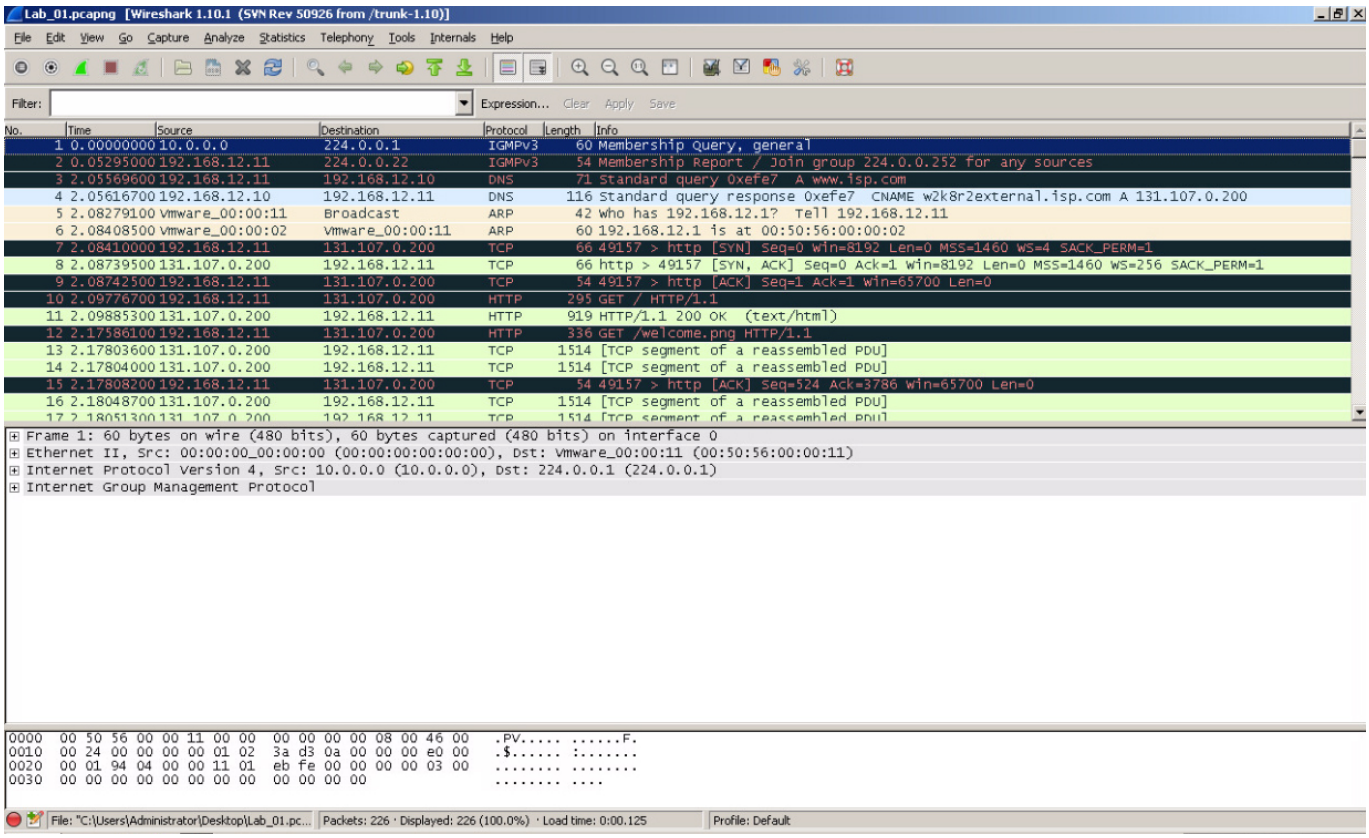

INTERNAL 192.168.12.11 WINDOWS SERVER

4. **Double-click** on the Lab_01 file on the desktop to open the Wireshark.



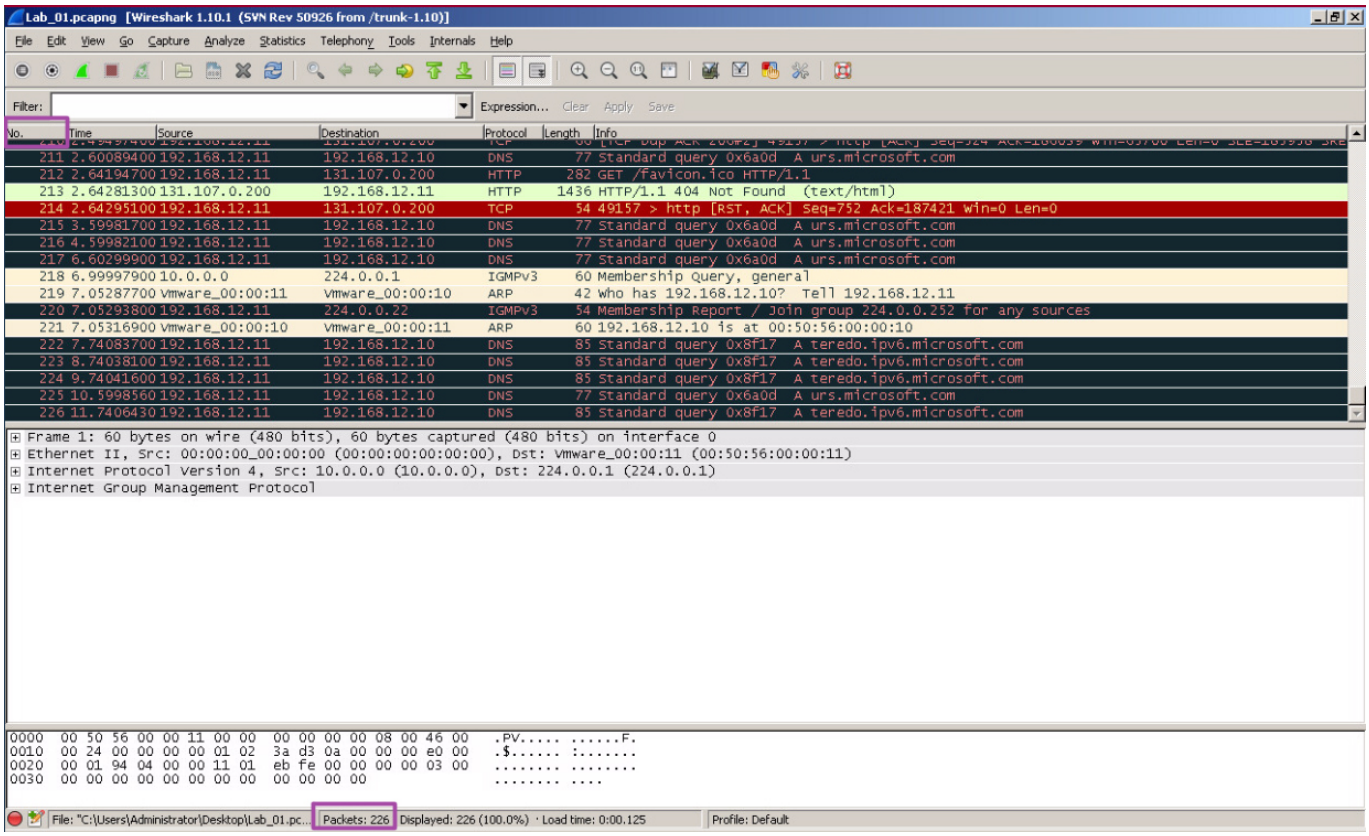

**LAB_01 FILE**

5. Once the file has opened, **take** a moment to get familiar with the Capture window. The top pane of the window shows the individual captured packets. The middle pane shows the details for the currently selected packet. The bottom pane shows the packet content.



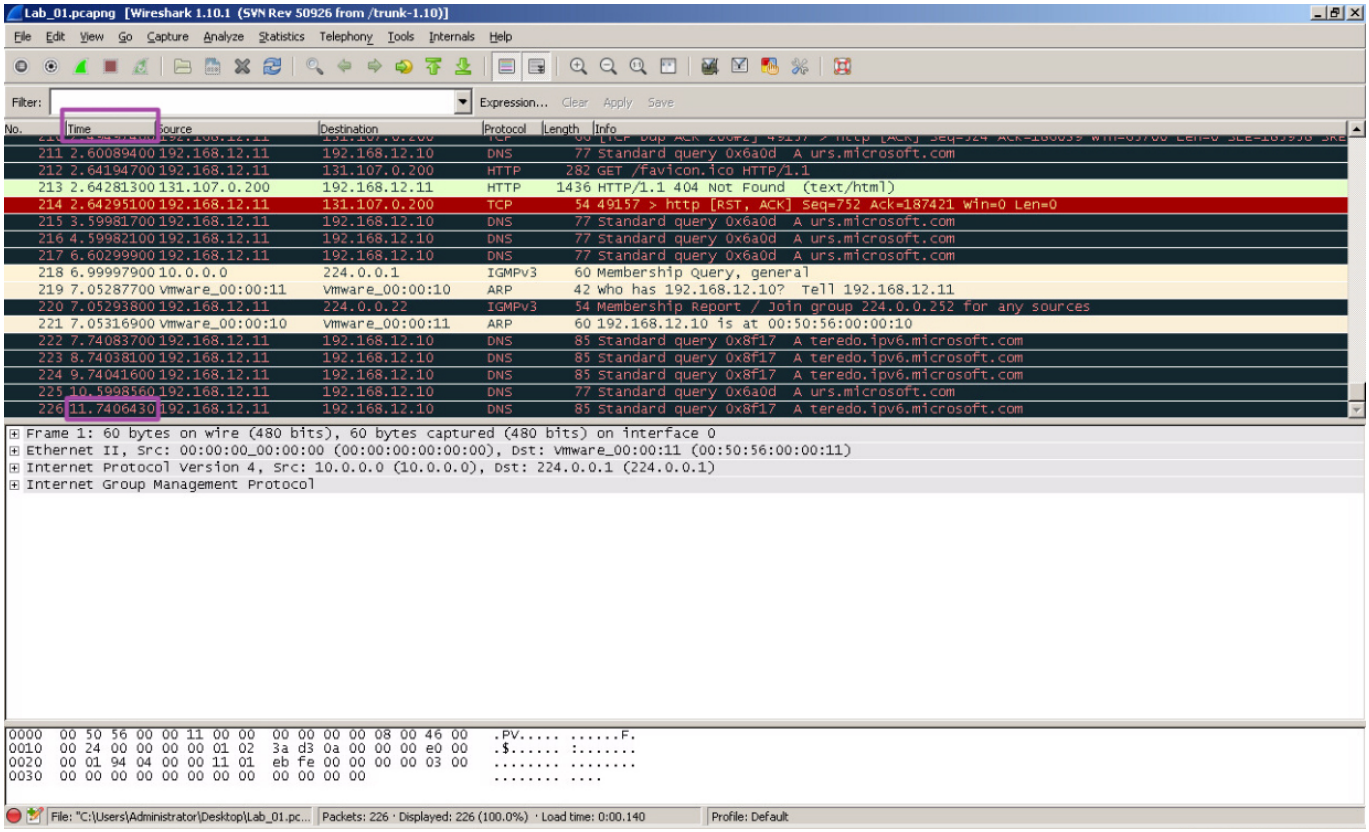

CAPTURE WINDOW

6. The first column in the captured packet pane is the packet number assigned in the order they were captured by the program. Scrolling through the list, you will notice there were a total of 226 packets captured in this example.



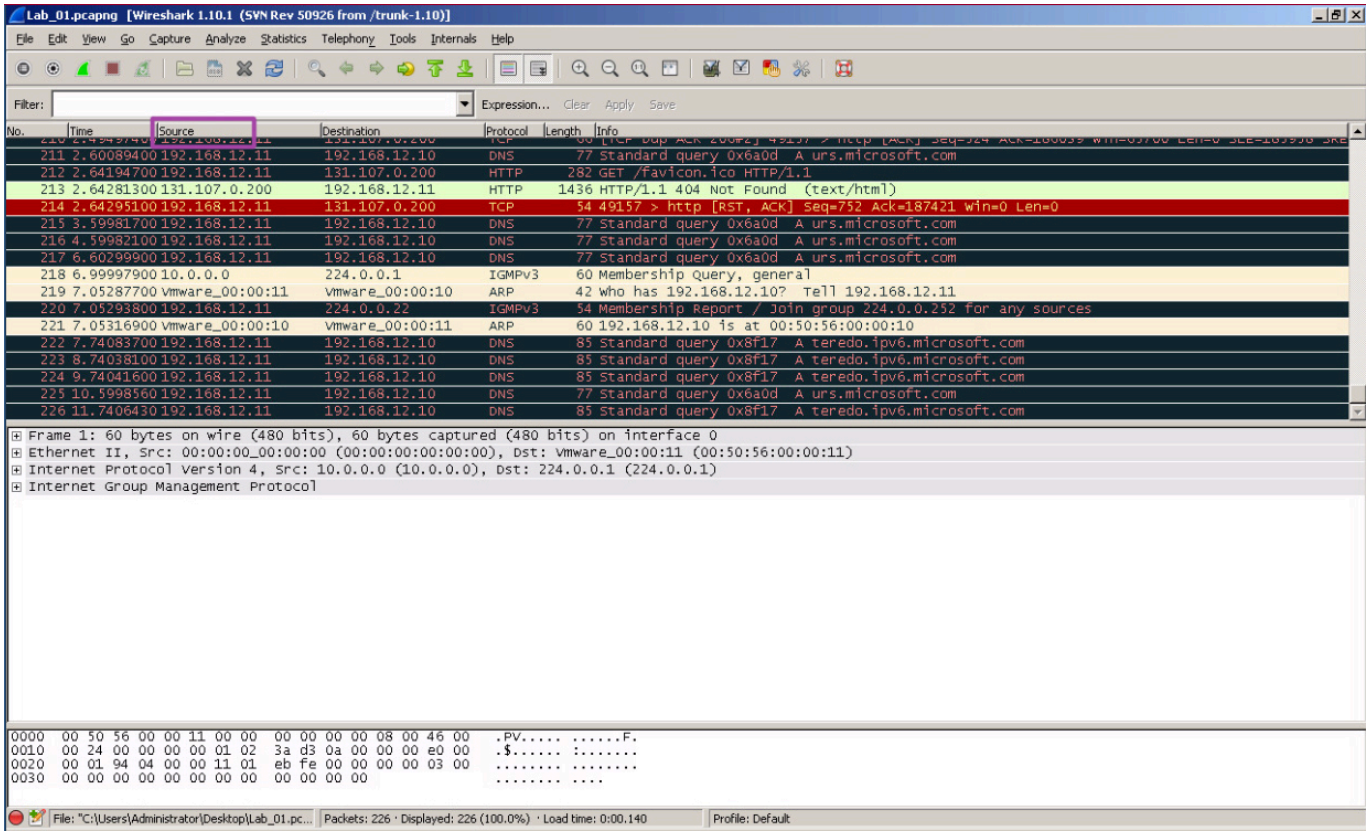

CAPTURE WINDOW

7. The second column shows the time at which the packet was captured in reference to when the capture was initiated. Scrolling through the list, you will notice the last packet was captured approximately 11.74 seconds after the capture was started.



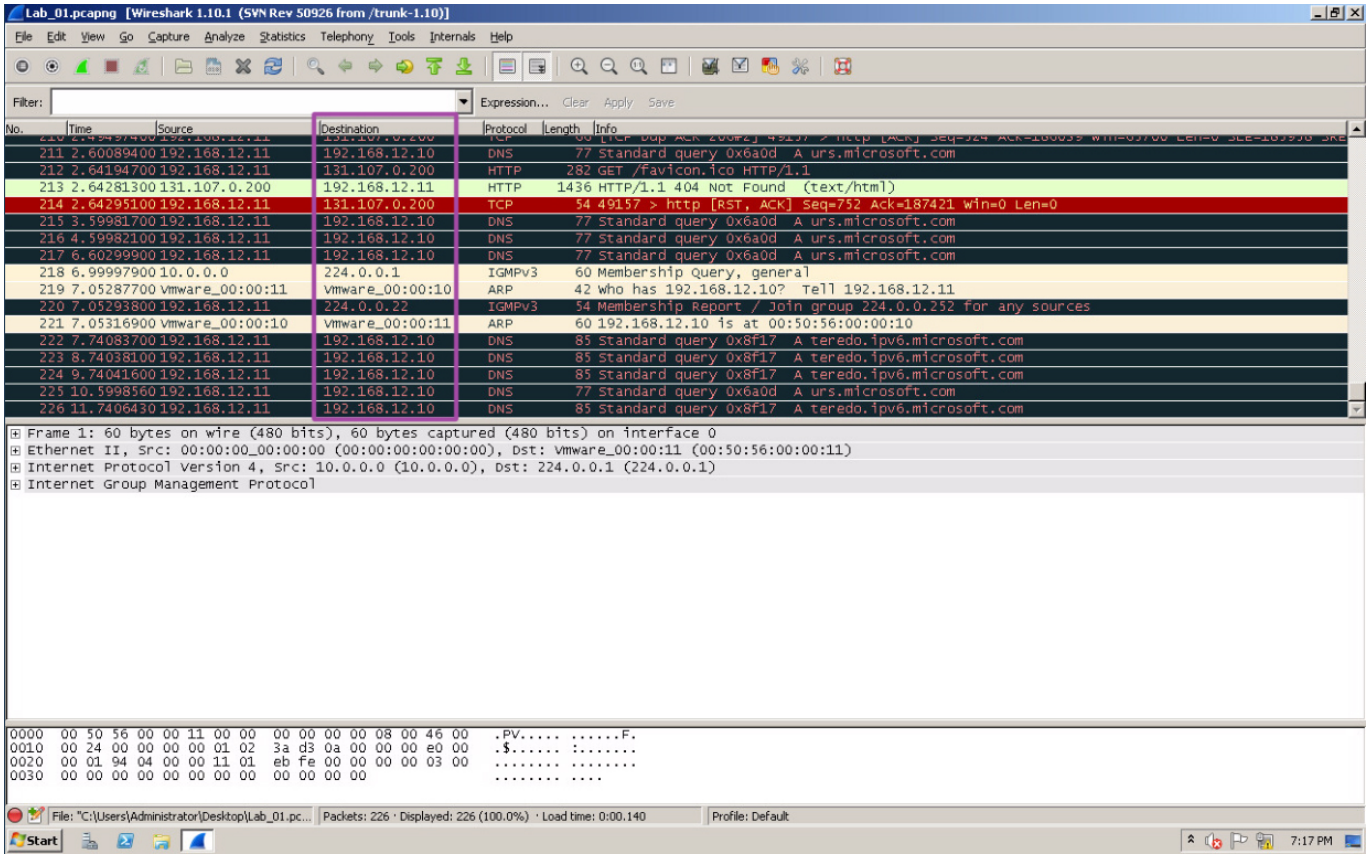

### CAPTURE WINDOW

8. The third column is the source IP address associated with the packet that was captured. The source is where the packet came from. Scrolling through the list, you will notice several examples of source IP addresses including some that you will work with in this lab. These addresses include 192.168.12.11, 192.168.12.10, and 131.107.0.200.



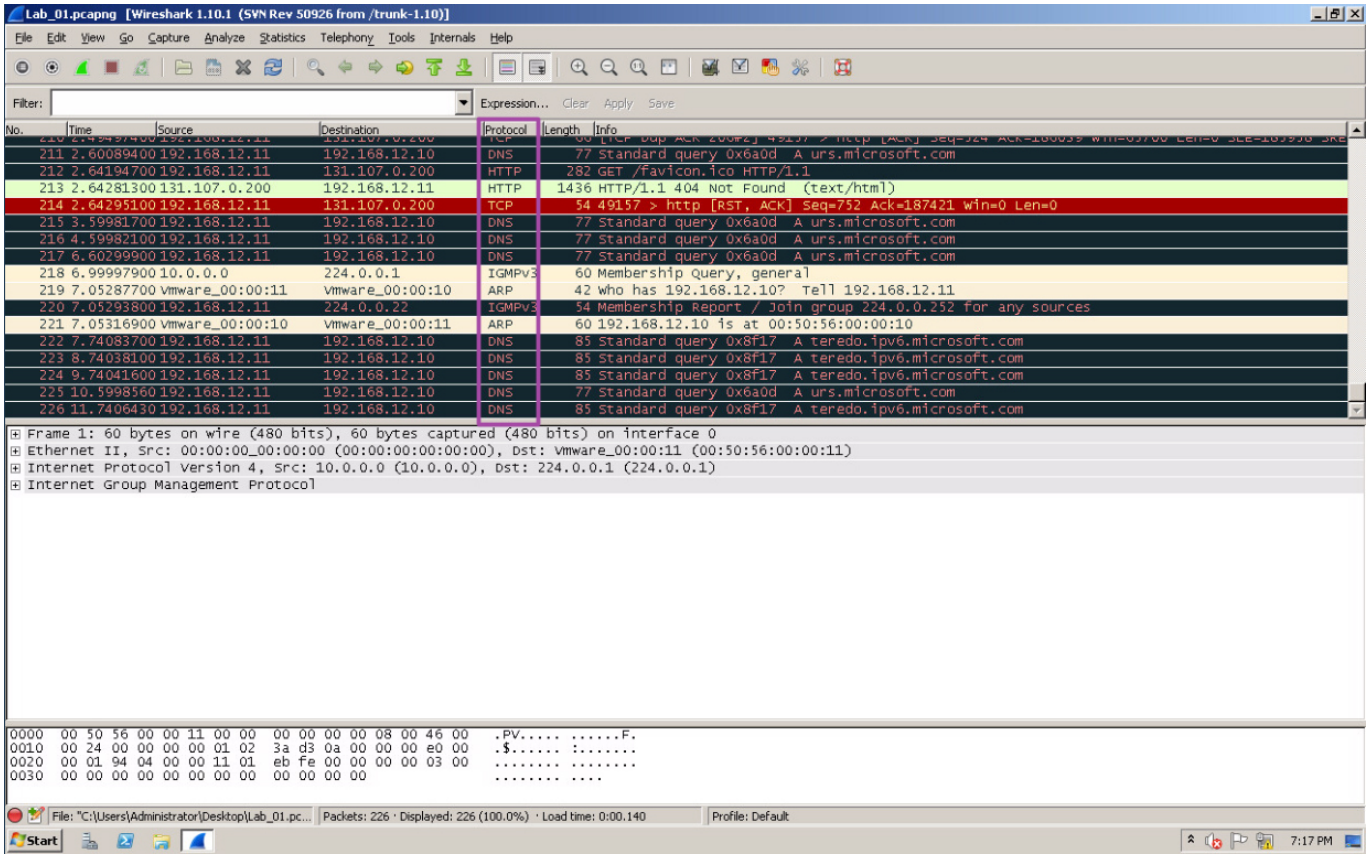

## CAPTURE WINDOW

9. The fourth column is the destination IP address associated with the packet that was captured. The destination is where the packet is going. Scrolling through the list, you will notice that many of the source addresses you just saw are also included in this column. This shows the two-way conversation between these machines.



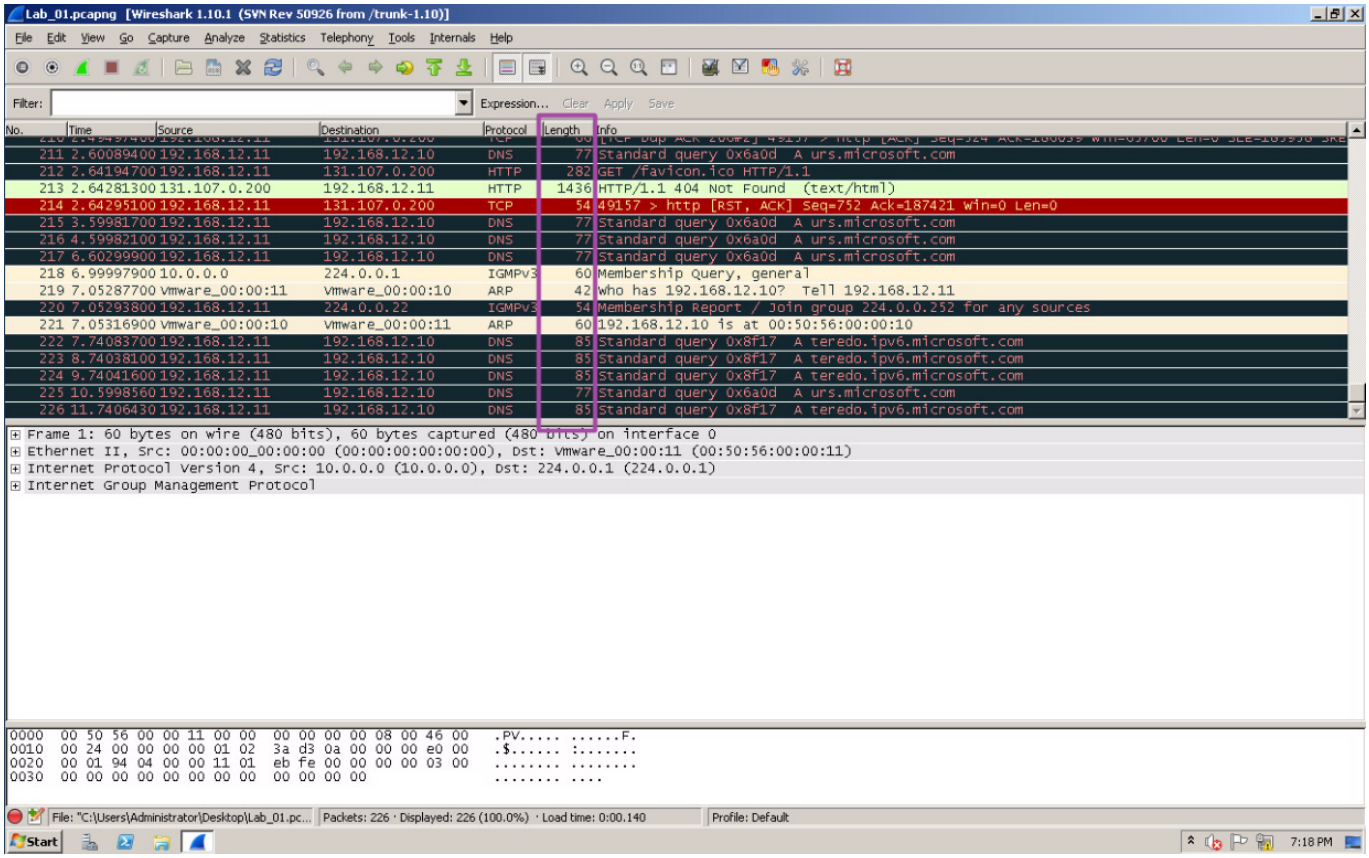
CAPTURE WINDOW

10. The fifth column indicates the protocol being used within the captured packet. Scrolling through the list, you will notice several protocols associated with this conversation, including HTTP, TCP, DNS, and ARP.



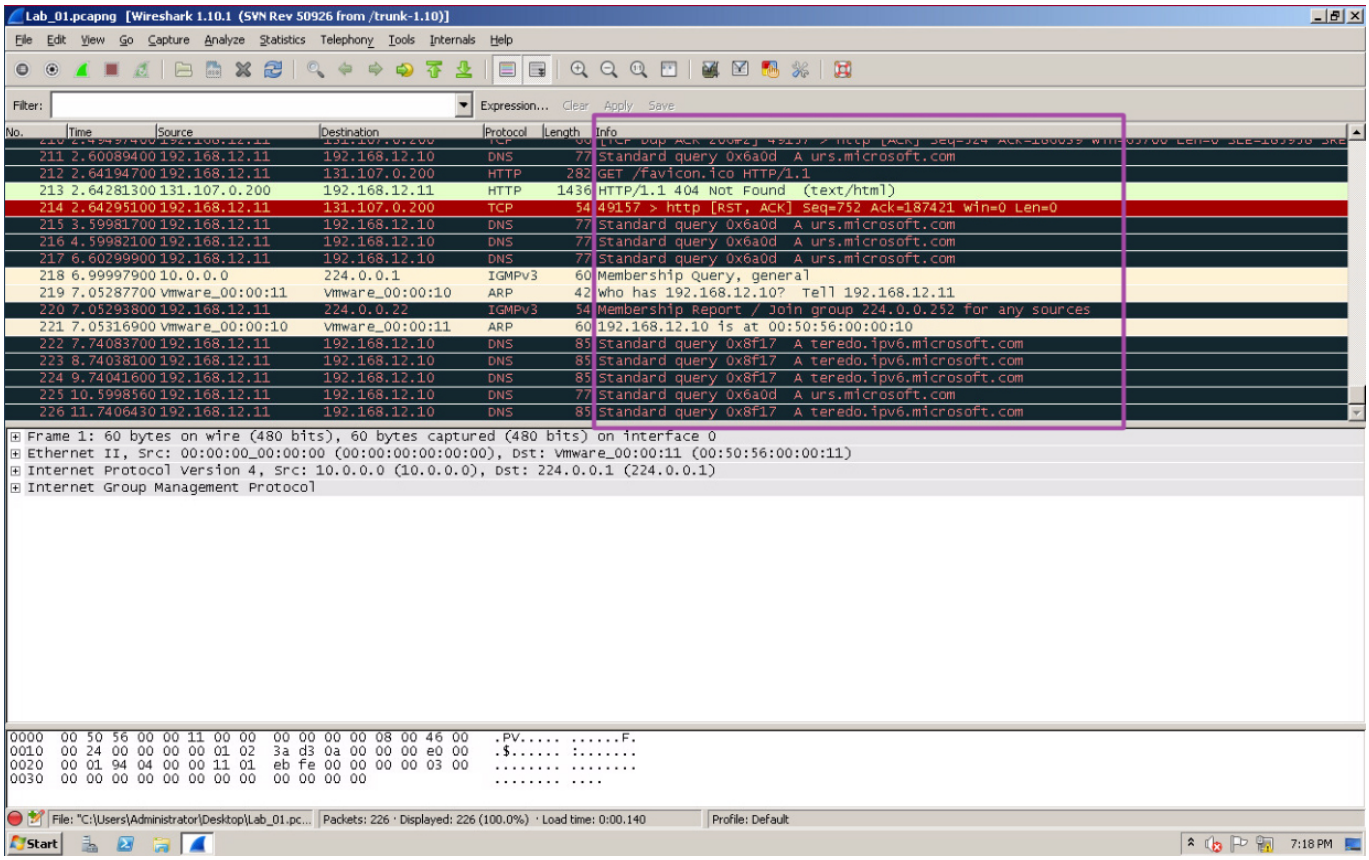

CAPTURE WINDOW

11. The sixth column is the length of the captured packet. Scrolling through the list, you will notice packets vary greatly in size. For example, DNS or ARP packets are relatively small, while several of the TCP packets are relatively large.



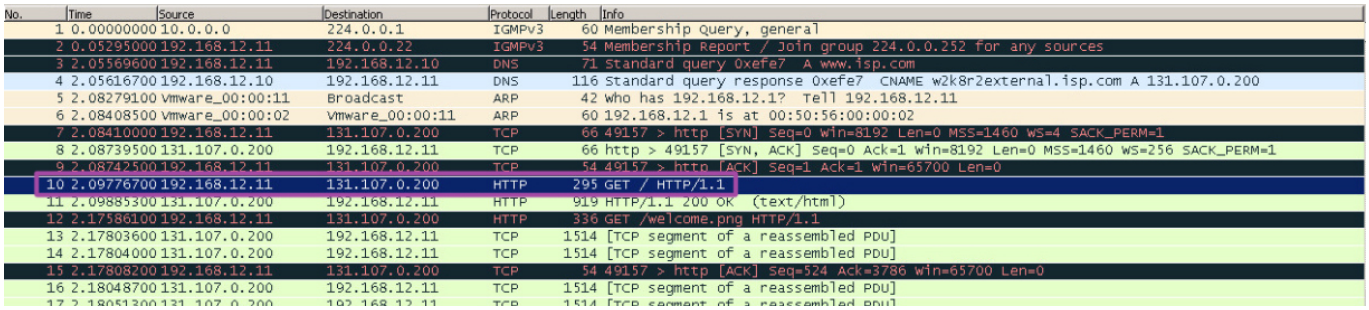

CAPTURE WINDOW

12. The seventh and final column gives you information about what is inside of the packet. Scrolling through the list, the information within packets will vary greatly.



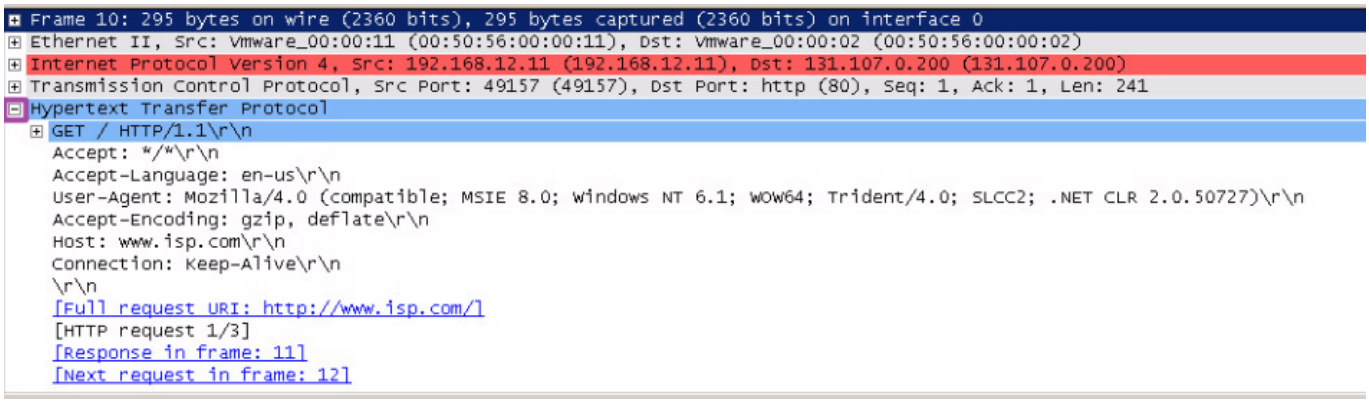

CAPTURE WINDOW

13. **Scroll** in the list until you see packet number 10. **Select** this packet by clicking on it in the top pane of the Capture window.



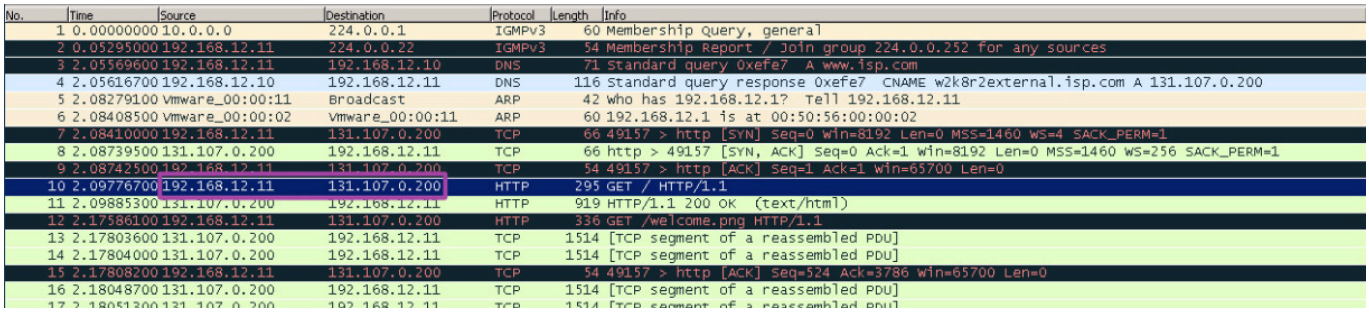

### PACKET NUMBER 10

14. In the middle pane of the Capture window, **expand** the + next to Hypertext Transfer Protocol.



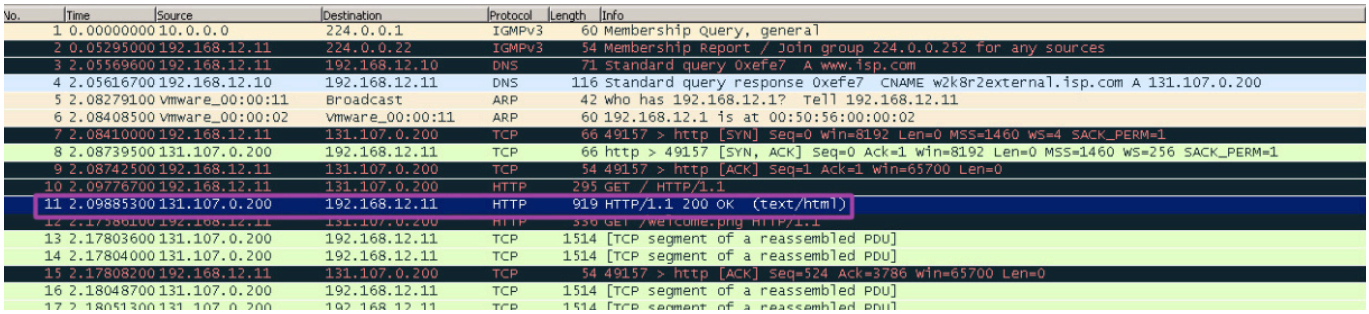

### HYPERTEXT TRANSFER PROTOCOL

15. Hypertext Transfer Protocol (HTTP) is one of the application layer protocols in the TCP/IP suite. What you are currently looking at is the initial request from the web client to the web server for the website http://www.isp.com. This can be identified by the line GET / HTTP/1.1. GET messages are used to request information from web servers. Referring back to the highlighted packet number 10, you can also use the source and destination IP address fields to see where the request is coming from and going to.




### SOURCE AND DESTINATION IP ADDRESS FIELDS

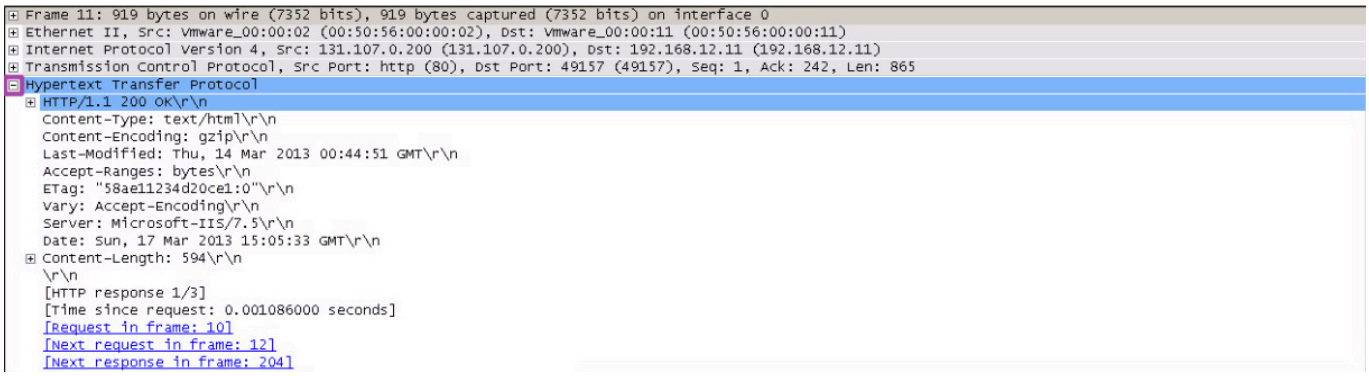
16. Now **select** packet number 11 in the top Capture window.



| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 1 | 0.00000000 | 10.0.0.0 | 224.0.0.1 | IGMPv3 | 60 | Membership Query, general |
| 2 | 0.05295000 | 192.168.12.11 | 224.0.0.22 | IGMPv3 | 54 | Membership Report / Join group 224.0.0.252 for any sources |
| 3 | 2.05569600 | 192.168.12.11 | 192.168.12.10 | DNS | 71 | Standard query 0xefe7 A www.isp.com |
| 4 | 2.05616700 | 192.168.12.10 | 192.168.12.11 | DNS | 116 | Standard query response 0xefe7 CNAME w2k8r2external.isp.com A 131.107.0.200 |
| 5 | 2.08279100 | Vmware_00:00:11 | Broadcast | ARP | 42 | Who has 192.168.12.1? Tell 192.168.12.11 |
| 6 | 2.08408500 | Vmware_00:00:02 | Vmware_00:00:11 | ARP | 60 | 192.168.12.1 is at 00:50:56:00:00:02 |
| 7 | 2.08410000 | 192.168.12.11 | 131.107.0.200 | TCP | 66 | 49157 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 |
| 8 | 2.08739500 | 131.107.0.200 | 192.168.12.11 | TCP | 66 | http > 49157 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 9 | 2.08742500 | 192.168.12.11 | 131.107.0.200 | TCP | 54 | 49157 > http [ACK] Seq=1 Ack=1 Win=65700 Len=0 |
| 10 | 2.09776700 | 192.168.12.11 | 131.107.0.200 | HTTP | 295 | GET / HTTP/1.1 |
| 11 | 2.09885300 | 131.107.0.200 | 192.168.12.11 | HTTP | 919 | HTTP/1.1 200 OK (text/html) |
| 12 | 2.17586100 | 192.168.12.11 | 131.107.0.200 | HTTP | 336 | GET /welcome.png HTTP/1.1 |
| 13 | 2.17803600 | 131.107.0.200 | 192.168.12.11 | TCP | 1514 | [TCP segment of a reassembled PDU] |
| 14 | 2.17804000 | 131.107.0.200 | 192.168.12.11 | TCP | 1514 | [TCP segment of a reassembled PDU] |
| 15 | 2.17808200 | 192.168.12.11 | 131.107.0.200 | TCP | 54 | 49157 > http [ACK] Seq=524 Ack=3786 Win=65700 Len=0 |
| 16 | 2.18048700 | 131.107.0.200 | 192.168.12.11 | TCP | 1514 | [TCP segment of a reassembled PDU] |
| 17 | 2.18051300 | 131.107.0.200 | 192.168.12.11 | TCP | 1514 | [TCP segment of a reassembled PDU] |

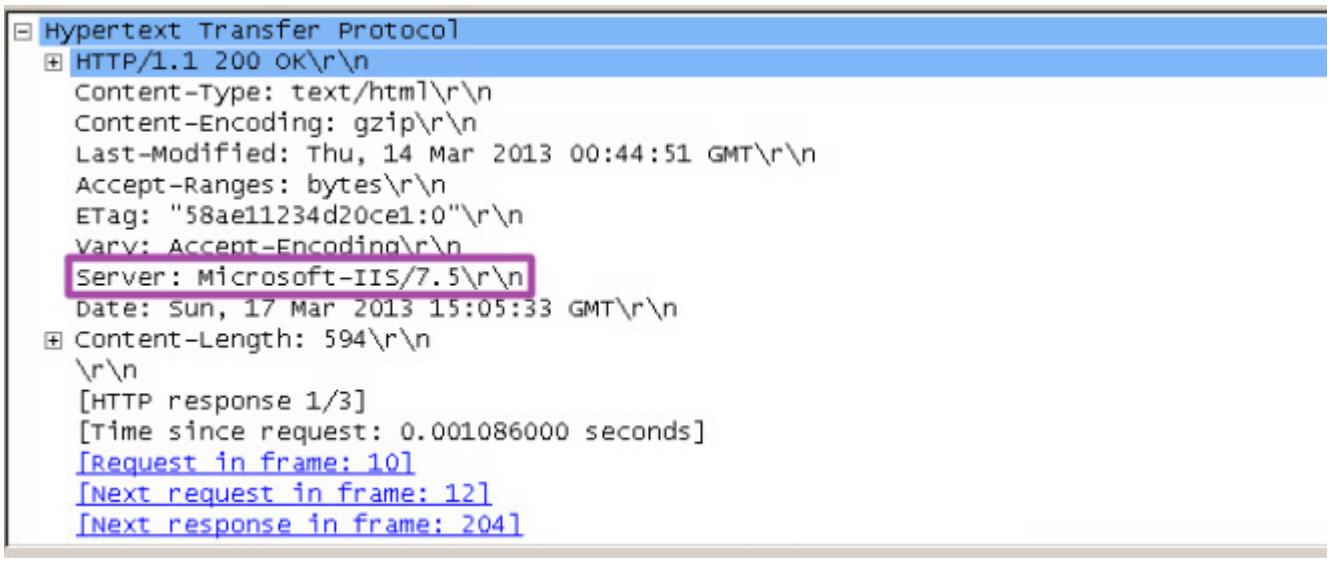
### PACKET NUMBER 11

17. In the middle pane of the capture window, **expand** the + next to Hypertext Transfer Protocol.




### HYPERTEXT TRANSFER PROTOCOL

18. What you are currently seeing is the initial response from the web server to the web client. **Look** for the line labeled Server. This line shows the service responding to the request on the server. In this example, the web server is running Microsoft Internet Information Services (IIS) version 7.5.




### WEB SERVER

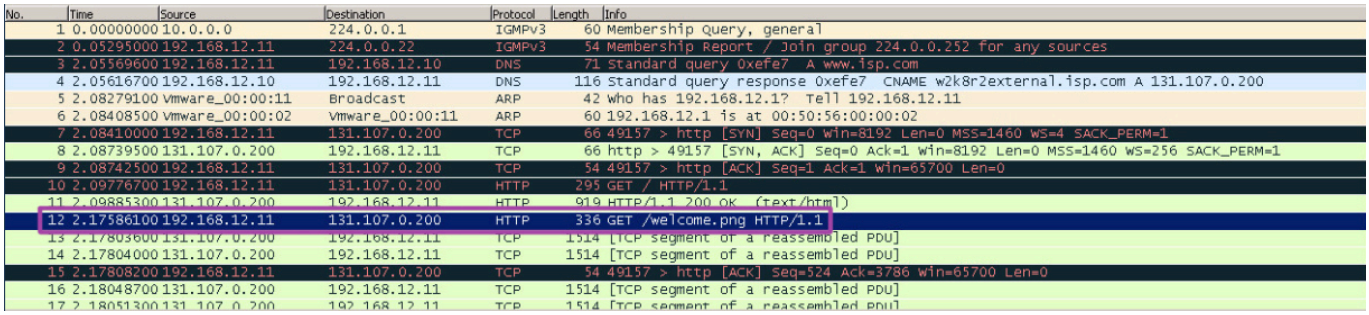
19. **Scroll down** and **expand** the + next to Line-based text data: text/html.


LINE-BASED TEXT DATA

20. As you scroll down, you are looking at the html code and text that make up the requested web page. This is interpreted by the web browser application on the client machine and the webpage is displayed. Scroll to the very bottom of the text and locate the line that begins with <a href=. Reading across, you will see the text img src="welcome.png." This line is a reference to an image that should be displayed on the webpage. The href reference before this is actually a link to a different website that you would be taken to if you clicked on the welcome.png picture on the webpage. The client will need to request this image before it can be displayed.

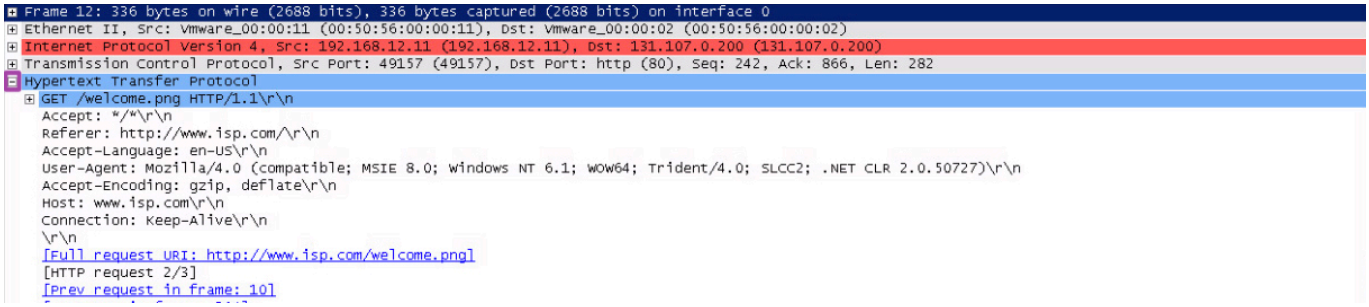

HREF SAMPLE HTML CODE

21. **Select** packet number 12 in the top capture window.



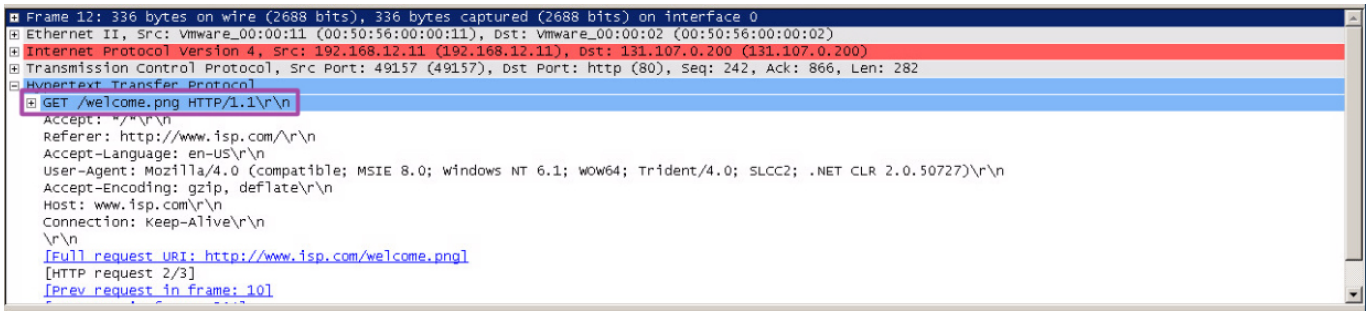

PACKET NUMBER 12

22. In the middle pane of the capture window, **expand** the + next to Hypertext Transfer Protocol.



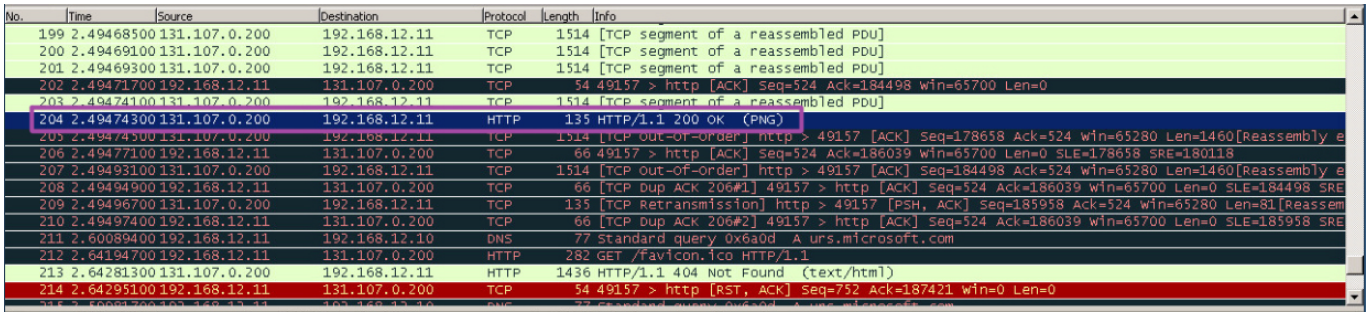

### HYPERTEXT TRANSFER PROTOCOL

23. **Notice** the line GET /welcome.png. This is the request from the web client to the web server to obtain the image named welcome.png. This image will be displayed on the requested web page.



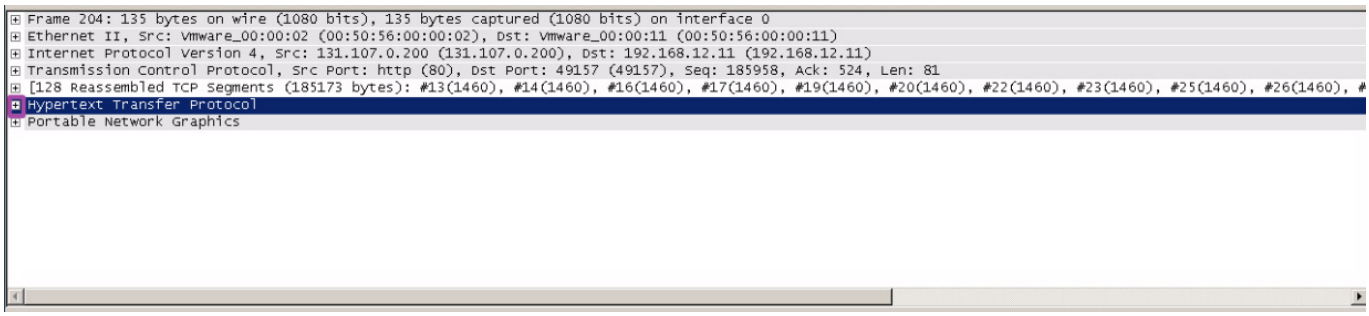

### GET /WELCOME.PNG LINE

24. **Scroll down** through the top capture window and **click** on packet number 204.




### PACKET NUMBER 204

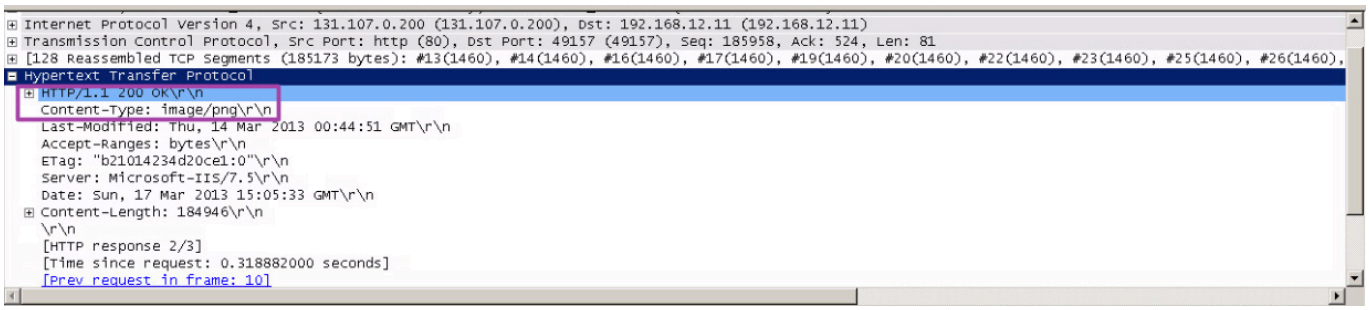
25. In the middle pane of the capture window, **expand** the + next to Hypertext Transfer Protocol.




### HYPERTEXT TRANSFER PROTOCOL

26. **Notice** the line HTTP/1.1 200 OK. This message signifies that the web server has processed the client request for the image and the image should have been sent to the client's web browser. (We will review the packets we skipped at a later time.) **Notice** the next line labeled Content-Type. This

also shows that a PNG formatted image was requested from the server.




**HTTP/1.1 200 OK**

# CONCLUSION:

HTTP is an application layer protocol of the OSI model. End-user applications, such as a web browser, use this protocol to send a request for required web based information and the server packages data that makes up the web page to respond to the request.

# DISCUSSION QUESTIONS:

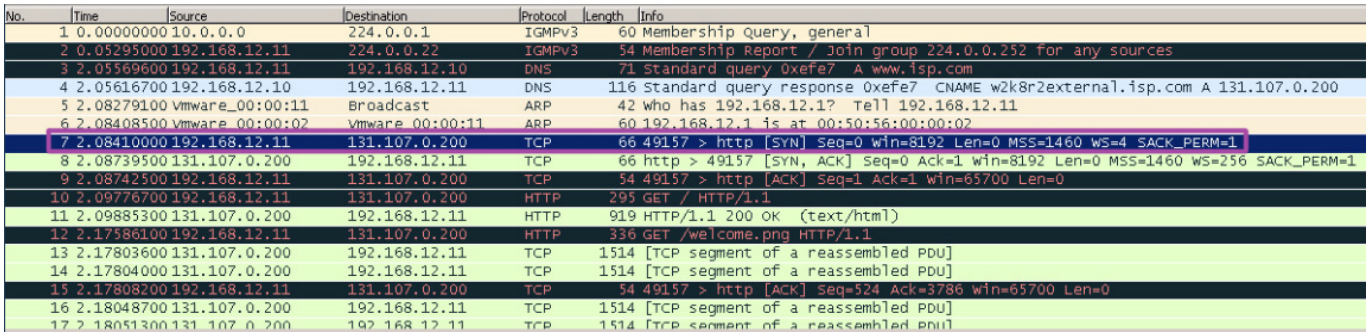
1. What is the PDU associated with the top three layers of the OSI model?
2. What HTTP message type is used to request data?
3. What HTTP message type was used to signify the image was successfully transferred to the client?

# Reviewing the Transport Layer

The OSI model has multiple protocols at the transport layer. In the TCP/IP model, there are two protocols that reside at the transport layer, TCP and UDP. TCP and UDP are the most widely referenced transport protocols in the OSI, and most of the TCP and UDP functions map to the OSI transport layer. TCP and UDP use port numbers to differentiate between application transmissions. IANA uses RFC 6335 to describe the procedures for assigning port numbers. The TCP protocol is responsible for connection-oriented data transmission. TCP conversations always start with a three-way handshake. This process prepares both the server providing the information and the client receiving the information for the communication. TCP also uses acknowledgments to verify data transmission. The UDP protocol is responsible for connectionless data transmission. UDP only sends data – it does not send acknowledgments to verify data transmission. This layer is also responsible for breaking down large data into smaller, more manageable pieces. This process for TCP is known as segmentation. With UDP, the more manageable pieces are called datagrams and have no sequencing information included. The PDU associated with the transport layer of the OSI model is a segment for TCP and datagram for UDP.

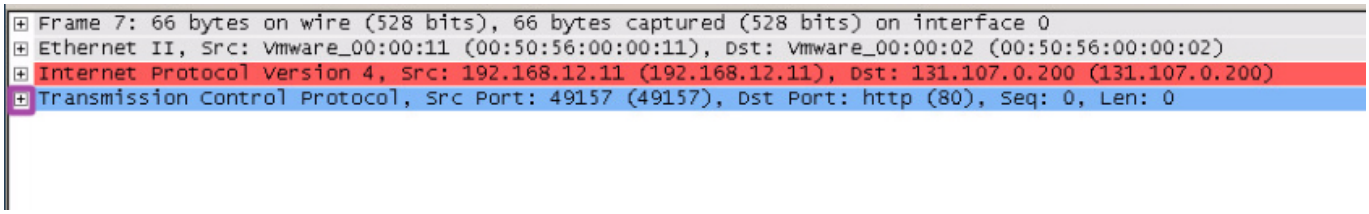
## Segment Protocol Data Unit

1. On the Windows Server machine, **scroll up** through the top capture window until you see frame number 7. **Select** this frame by clicking on it in the top capture window.



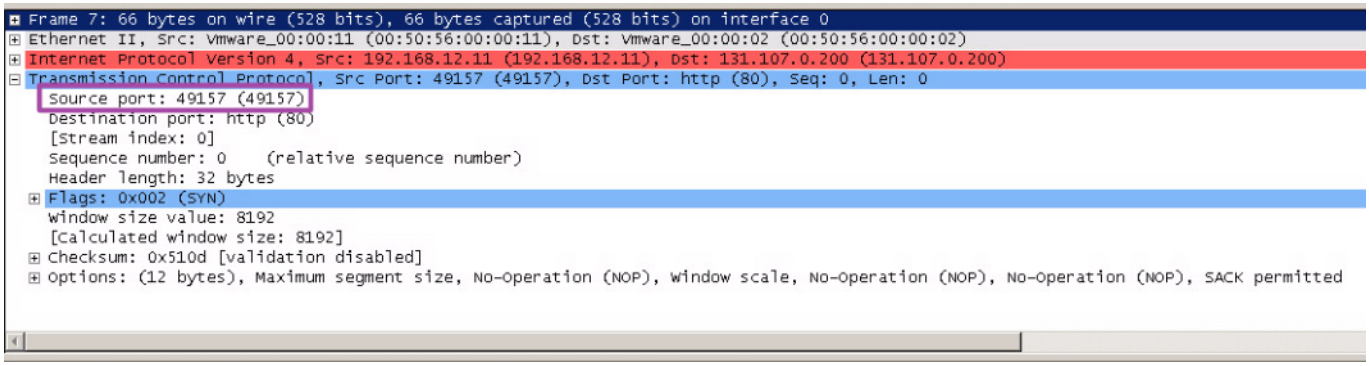

### FRAME NUMBER 7

2. In the middle pane of the capture window, **expand** the + next to Transmission Control Protocol.



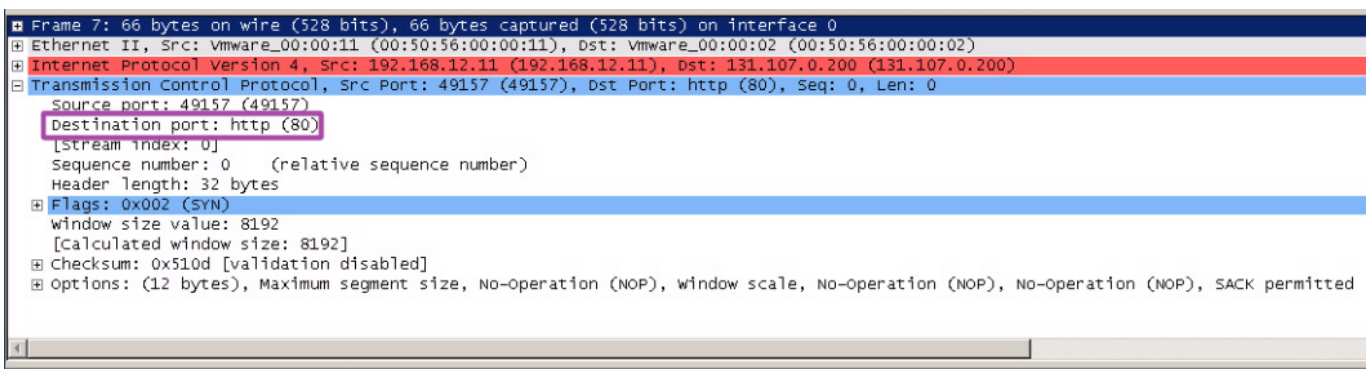

### TRANSMISSION CONTROL PROTOCOL

3. **Notice** the line Source Port. This port is a randomly generated number between 49152 and 65535 that the requesting client will use to keep track of this web page request. This range of ports is known as Dynamic Ports.



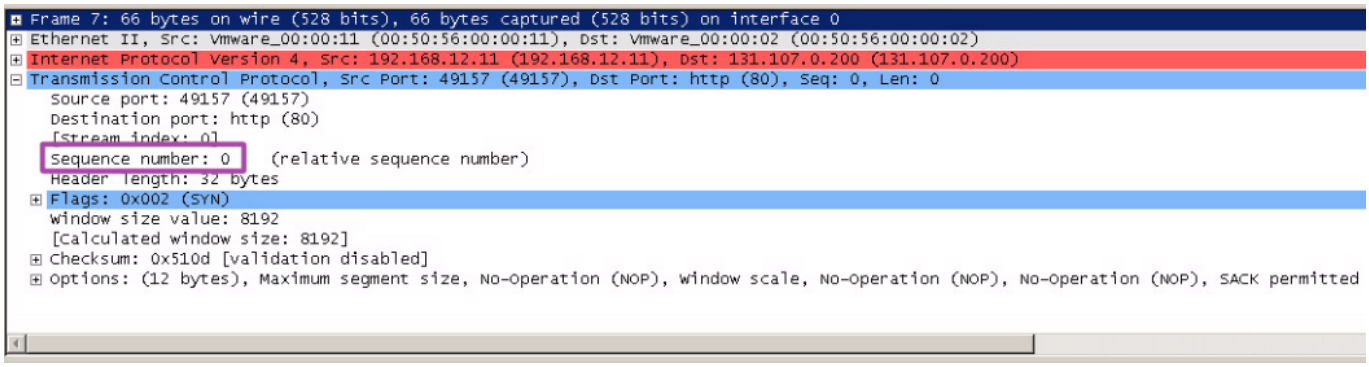

### SOURCE PORT

4. **Notice** the line Destination Port. Port 80, the destination port of this packet, is assigned by IANA specifically for the HTTP protocol. Ports that fall into the range 0-1023 are known as System Ports. Some texts also refer to this range as Well Known Ports. These ports are assigned to specific applications allowing the receiving server to identify the application. In this example port 80 indicates that the web server application needs to respond to the request.



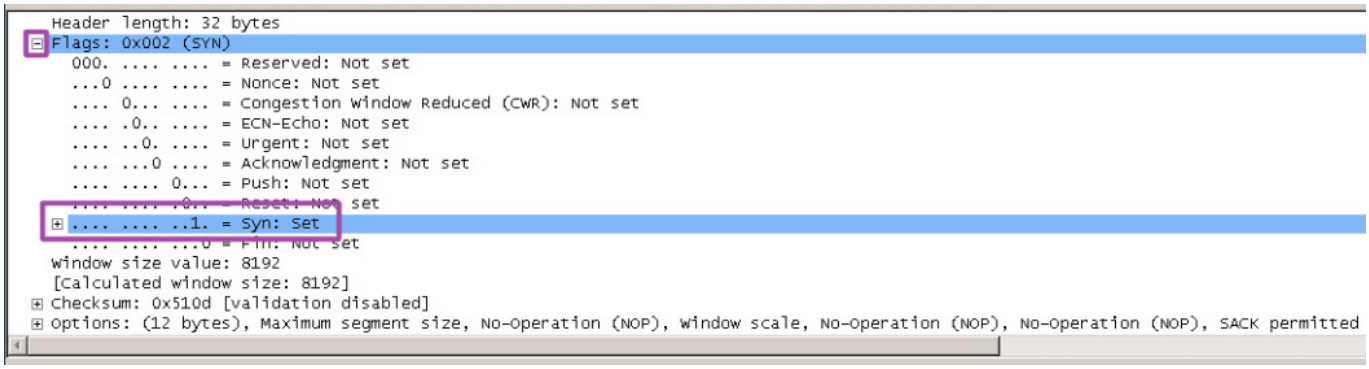
### DESTINATION PORT

5. **Notice** the line Sequence Number. Sequence numbers are used to keep all of the TCP segments in the correct order. The first segment in the TCP three-way handshake is always assigned sequence number 0 in a default Wireshark configuration. This segment is called the SYN segment.



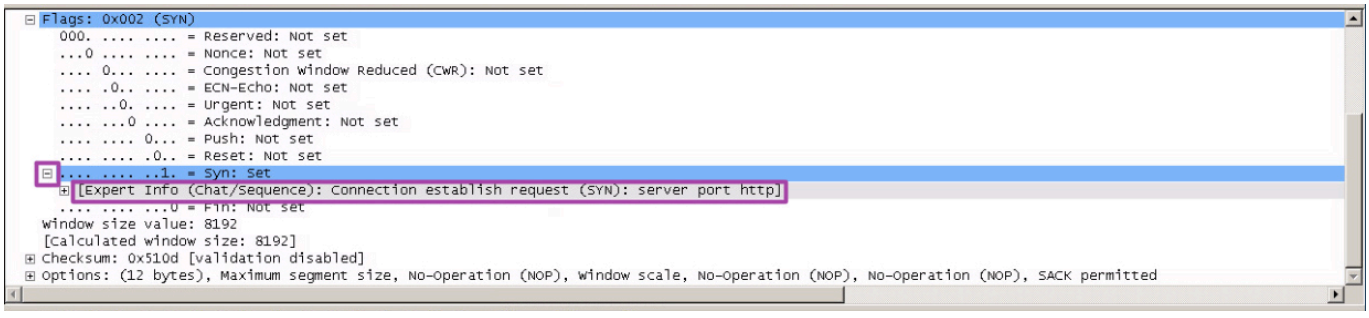

### SEQUENCE NUMBER

6. **Expand** the + next to Flags. Flags are used to set certain options available to the segment. In this example, there is one flag set – the Syn flag. This can be observed by noting the 1 bit is set in this field, while all others have the bit set to 0.



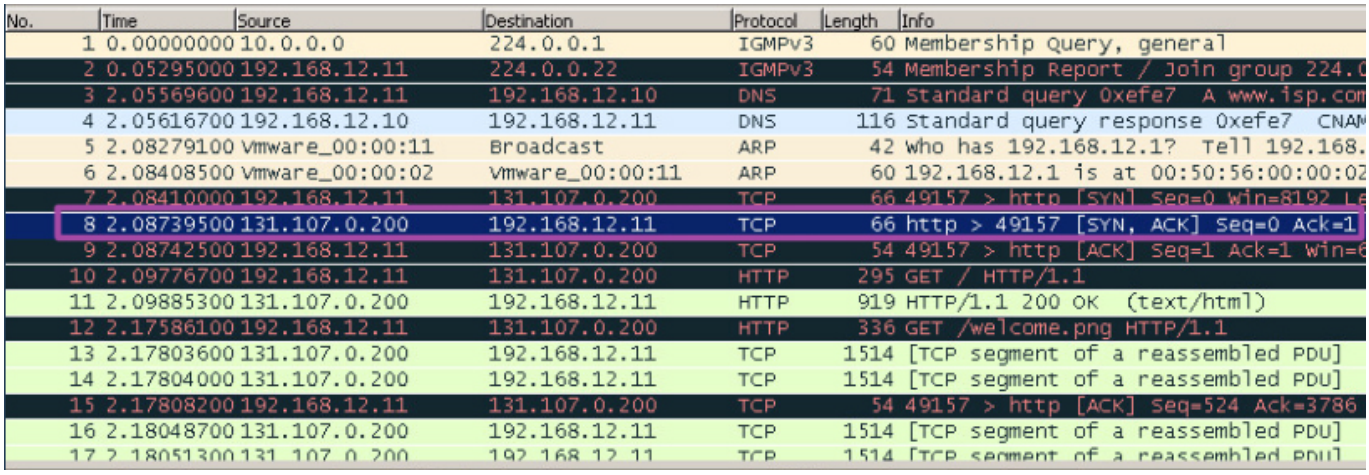

### FLAGS

7. **Expand** the + next to the line Syn: Set. Notice the line Expert Info. The purpose of this flag is explained. The SYN segment is used to request a connection from the client to the server.




### EXPERT INFO

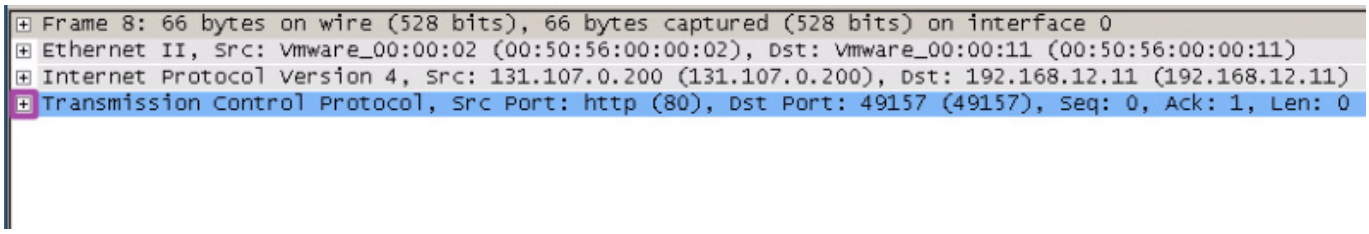
8. In the top capture window, **select** packet number 8.



| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 1 | 0.00000000 | 10.0.0.0 | 224.0.0.1 | IGMPv3 | 60 | Membership Query, general |
| 2 | 0.05295000 | 192.168.12.11 | 224.0.0.22 | IGMPv3 | 54 | Membership Report / Join group 224.0 |
| 3 | 2.05569600 | 192.168.12.11 | 192.168.12.10 | DNS | 71 | Standard query 0xefe7 A www.isp.com |
| 4 | 2.05616700 | 192.168.12.10 | 192.168.12.11 | DNS | 116 | Standard query response 0xefe7 CNAM |
| 5 | 2.08279100 | Vmware_00:00:11 | Broadcast | ARP | 42 | Who has 192.168.12.1? Tell 192.168. |
| 6 | 2.08408500 | Vmware_00:00:02 | Vmware_00:00:11 | ARP | 60 | 192.168.12.1 is at 00:50:56:00:00:02 |
| 7 | 2.08410000 | 192.168.12.11 | 131.107.0.200 | TCP | 66 | 49157 > http [SYN] Seq=0 Win=8192 Le |
| 8 | 2.08739500 | 131.107.0.200 | 192.168.12.11 | TCP | 66 | http > 49157 [SYN, ACK] Seq=0 Ack=1 |
| 9 | 2.08742500 | 192.168.12.11 | 131.107.0.200 | TCP | 54 | 49157 > http [ACK] Seq=1 Ack=1 Win=6 |
| 10 | 2.09776700 | 192.168.12.11 | 131.107.0.200 | HTTP | 295 | GET / HTTP/1.1 |
| 11 | 2.09885300 | 131.107.0.200 | 192.168.12.11 | HTTP | 919 | HTTP/1.1 200 OK (text/html) |
| 12 | 2.17586100 | 192.168.12.11 | 131.107.0.200 | HTTP | 336 | GET /welcome.png HTTP/1.1 |
| 13 | 2.17803600 | 131.107.0.200 | 192.168.12.11 | TCP | 1514 | [TCP segment of a reassembled PDU] |
| 14 | 2.17804000 | 131.107.0.200 | 192.168.12.11 | TCP | 1514 | [TCP segment of a reassembled PDU] |
| 15 | 2.17808200 | 192.168.12.11 | 131.107.0.200 | TCP | 54 | 49157 > http [ACK] Seq=524 Ack=3786 |
| 16 | 2.18048700 | 131.107.0.200 | 192.168.12.11 | TCP | 1514 | [TCP segment of a reassembled PDU] |
| 17 | 2.18051300 | 131.107.0.200 | 192.168.12.11 | TCP | 1514 | [TCP segment of a reassembled PDU] |

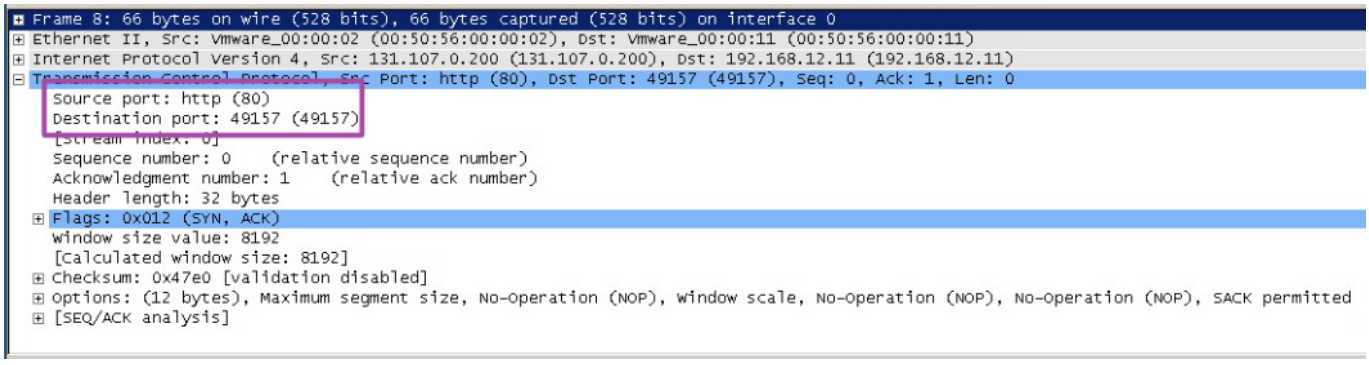
### PACKET NUMBER

9. In the middle pane of the capture window, **expand** the + next to Transmission Control Protocol.



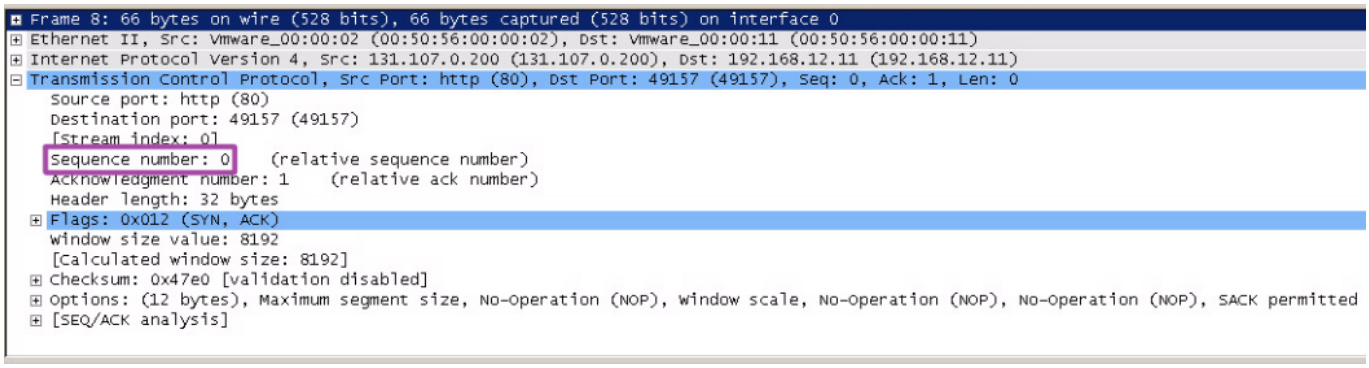

### TRANSMISSION CONTROL PROTOCOL

10. **Notice** the lines Source port and Destination port. You will see that the same port numbers are being used, but they have now changed positions. This is because this segment is a response from the server hosting the web page to the web client that requested the webpage. Because the application receiving the response is also a web-based application, the port number indicates the HTTP protocol in the returning packet to alert the client to use the web browser.



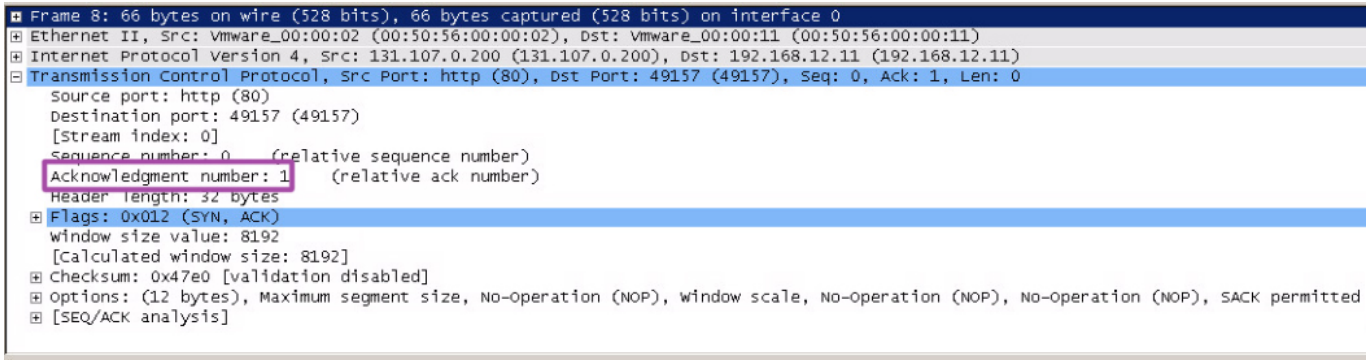

### SOURCE AND DESTINATION PORTS

11. **Notice** the line Sequence Number. Since this is the first segment coming from the server (and the second part of the three-way handshake), this sequence number is also set to 0.



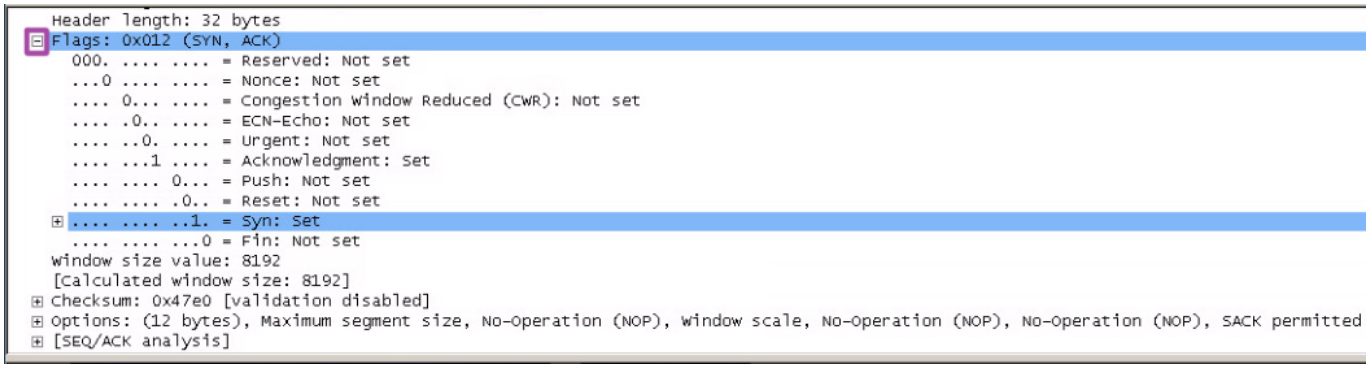

### SEQUENCE NUMBER

12. **Notice** the line Acknowledgment number. The TCP protocol uses acknowledgment numbers to indicate to the client that it has received its request and is responding to that request. The client in turn needs to use that acknowledgement number as the next sequence number because the server expects to see in the conversation. With the exception of the three-way handshake, acknowledgments are not sent for each segment. Instead, they are sent at periodic intervals set by a sliding window. This allows for greater efficiency since a large group of segments can be acknowledged at the same time. In this part of the three-way handshake, the acknowledgment number is 1.



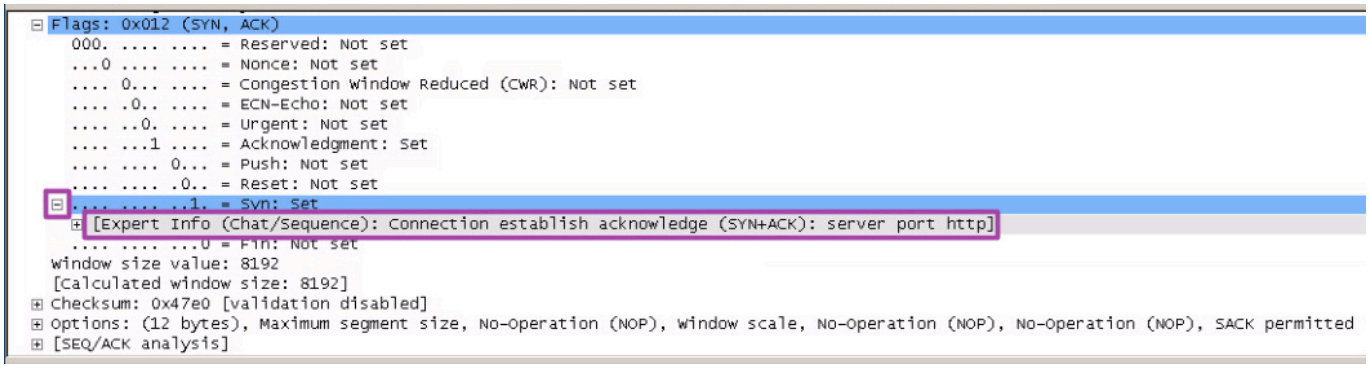

### ACKNOWLEDGMENT NUMBER

13. **Expand** the + next to Flags. Flags are used to set certain options available to the segment. In this example, there are two flags set – the Acknowledgment flag and the Syn flag.



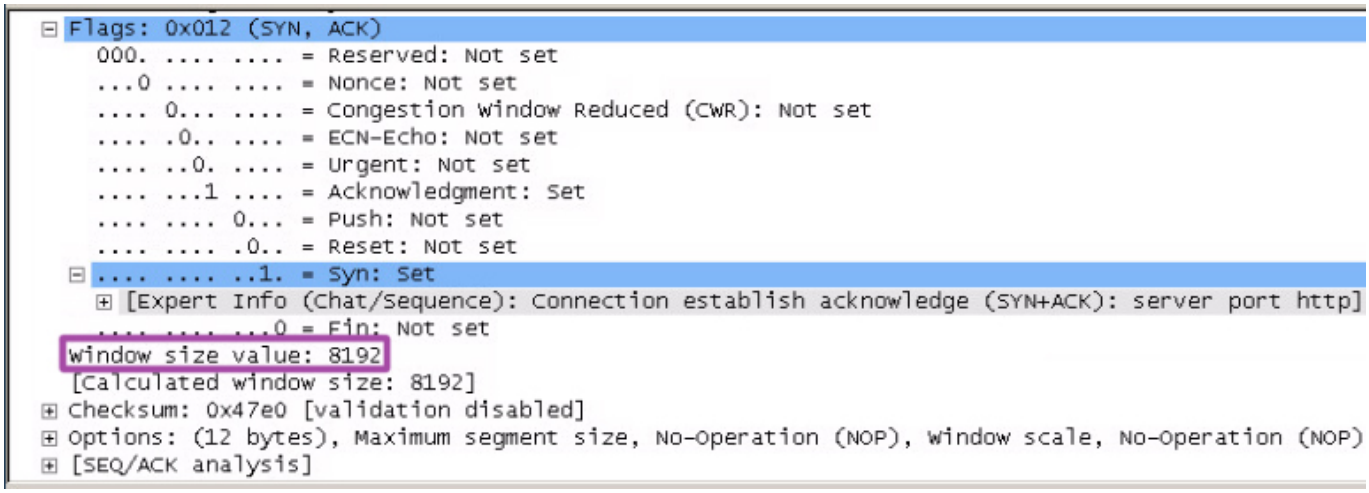

### FLAGS

14. **Expand** the + next to the line Syn: Set. **Notice** the line Expert Info. The purpose of this flag is explained. The SYN+ACK segment is used to acknowledge the request for a connection from the client to the server.



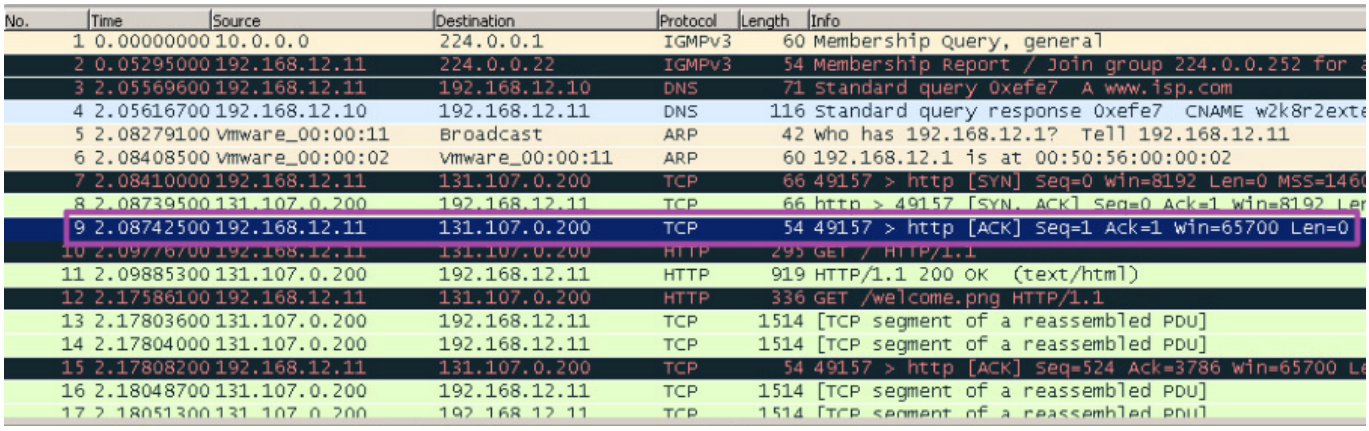

### SYN+ACK SEGMENT

15. **Notice** the line Window size value. This is the number of segments that will be sent before an acknowledgment is expected. It is called a sliding window because this value can change based on varying network conditions.



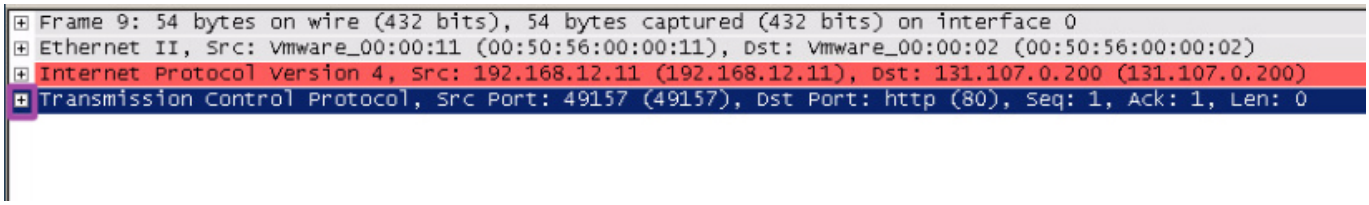

### WINDOWS SIZE VALUE

16. In the top capture window, **select** packet number 9 by clicking on it.



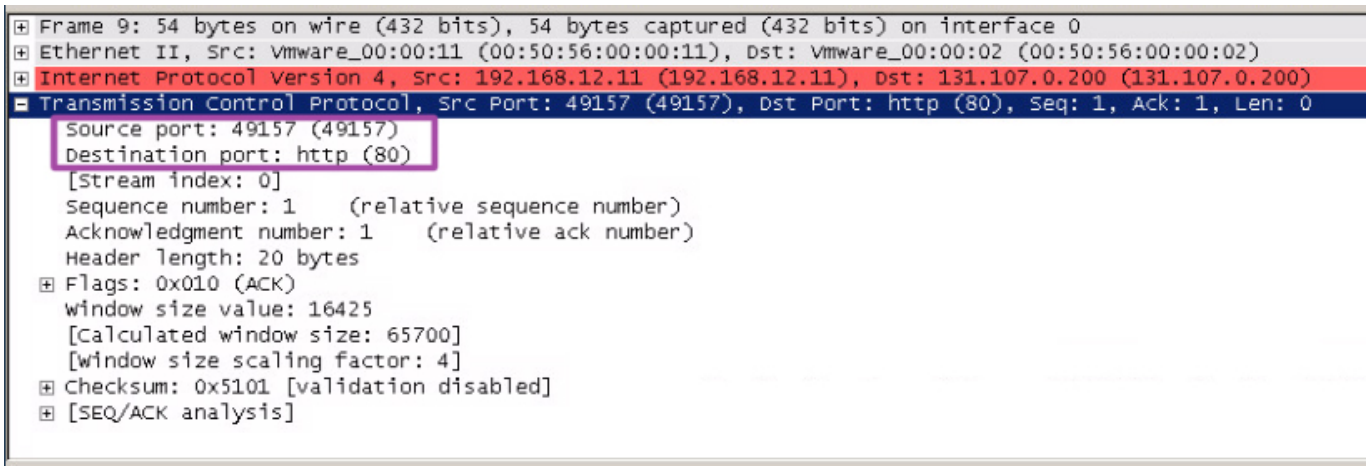

### PACKET NUMBER 9

17. In the middle pane of the capture window, **expand** the + next to Transmission Control Protocol.



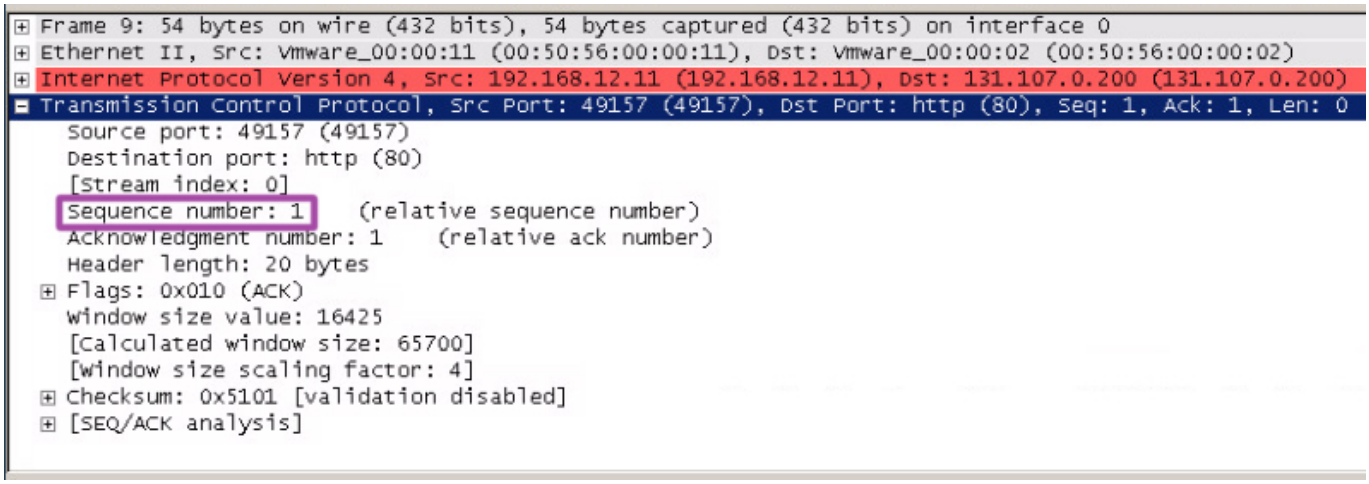

### TRANSMISSION CONTROL PROTOCOL

18. **Notice** the lines Source port and Destination port. You will see that the port numbers have returned to their original configuration. This is because this segment is the final response of the three-way handshake from the client to the server hosting the web page.



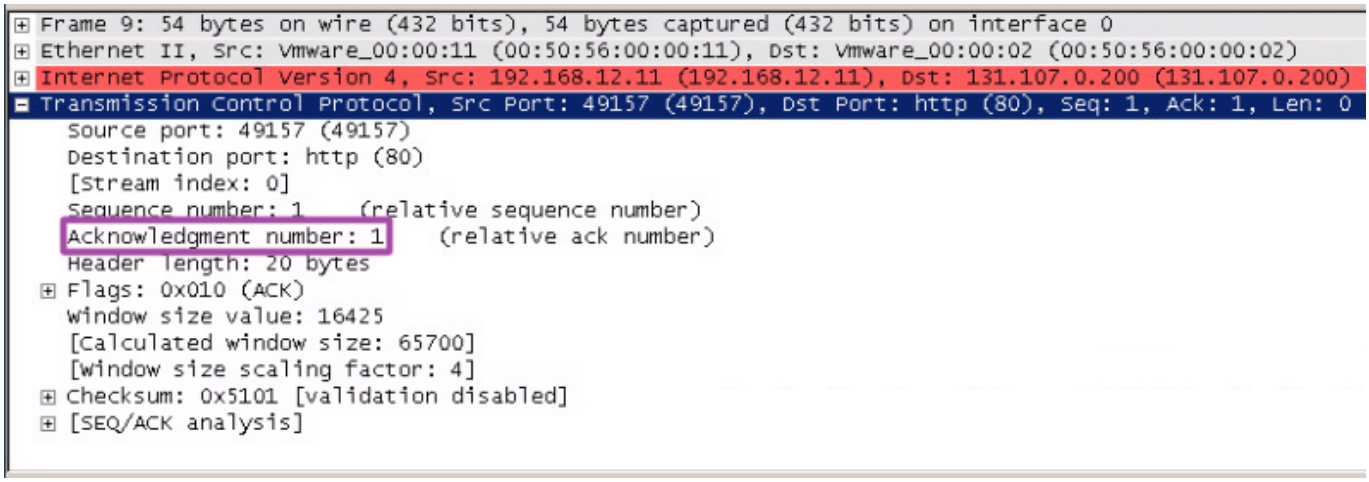

### SOURCE AND DESTINATION PORTS

19. **Notice** the line Sequence Number. Since this is the second segment coming from the client (and the third part of the three-way handshake), this sequence number is set to 1. This number also matches the acknowledgment number from the previous segment because the server is expecting to see segment number 1 next.



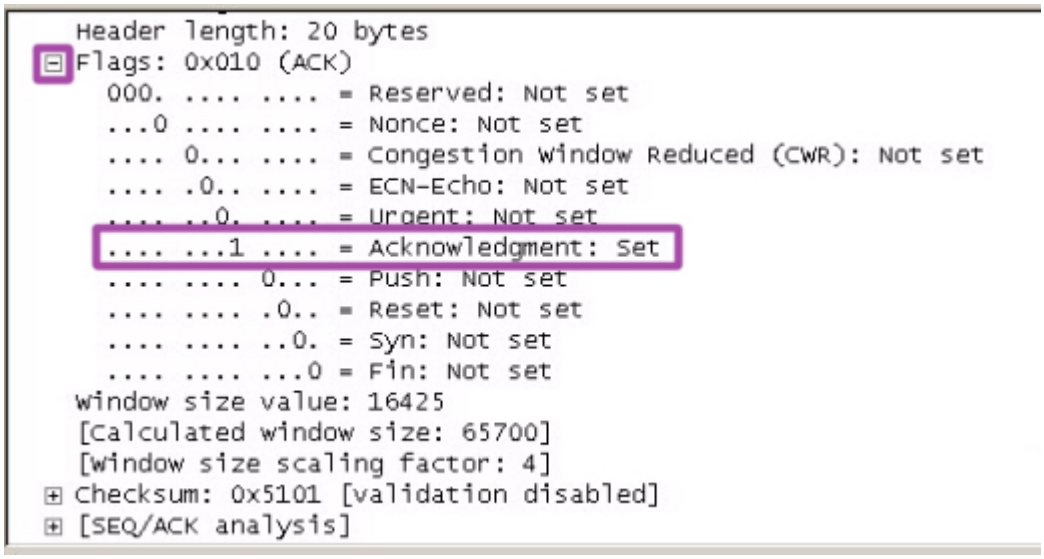

### SEQUENCE NUMBER

20. **Notice** the line Acknowledgment number. This number is also set to 1. The client is telling the web server that is did receive the initial segment and that it is expecting to see segment number 1 next.



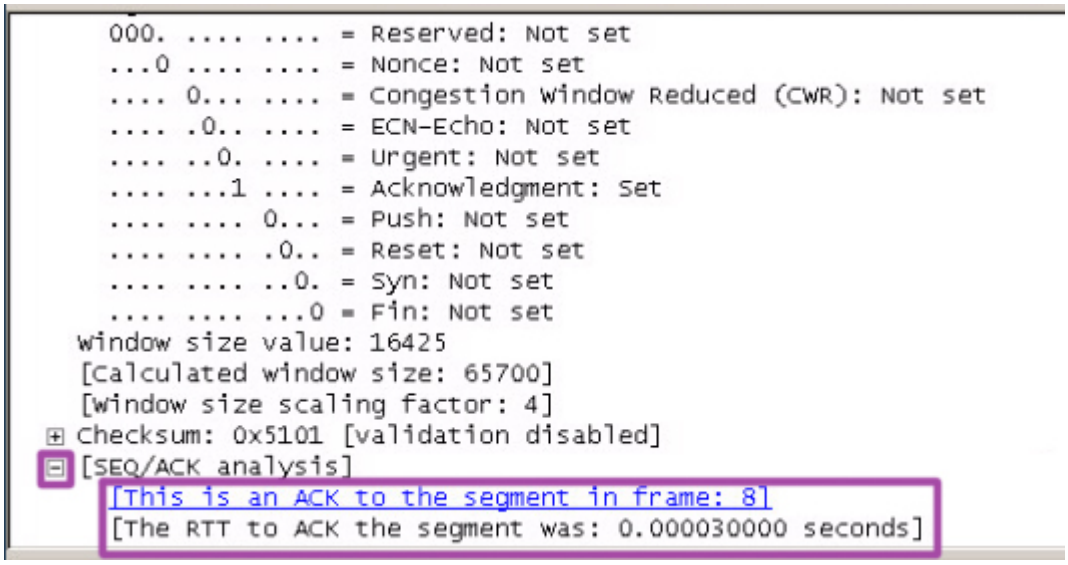

### ACKNOWLEDGMENT NUMBER

21. **Expand** the + next to Flags. Flags are used to set certain options available to the segment. In this example, there is only one flag set – the Acknowledgment flag.



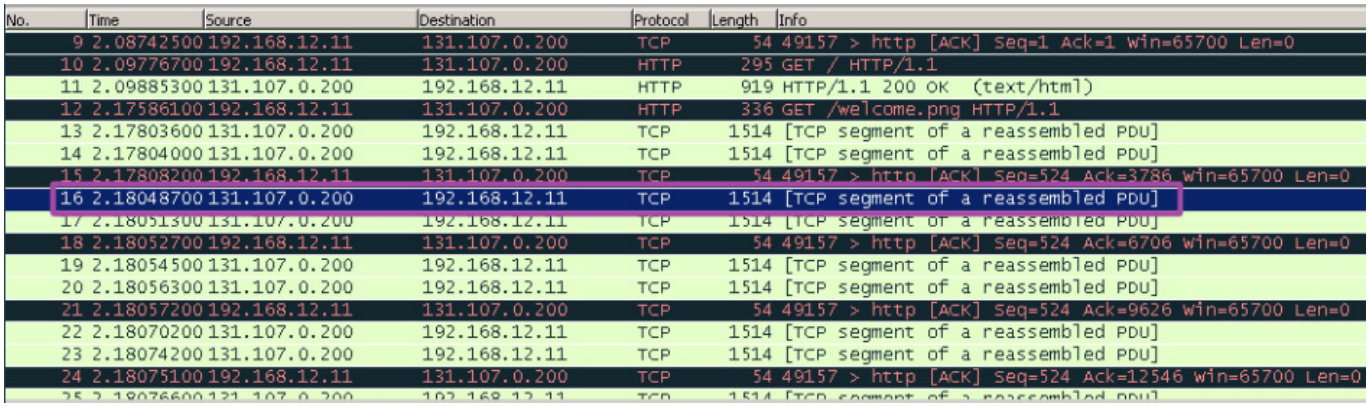

### ACKNOWLEDGMENT FLAG

22. **Scroll down** and **expand** the + next to the line SEQ/ACK analysis. This section actually tells you that this segment is an acknowledgment to the segment in frame number 8. It also includes the Round Trip Time, or how long it took for the acknowledgment to arrive. Once the TCP three-way handshake is complete, data transmission can begin. (These are the segments we skipped earlier.) These packets actually contain the data that makes up the picture on the web page.



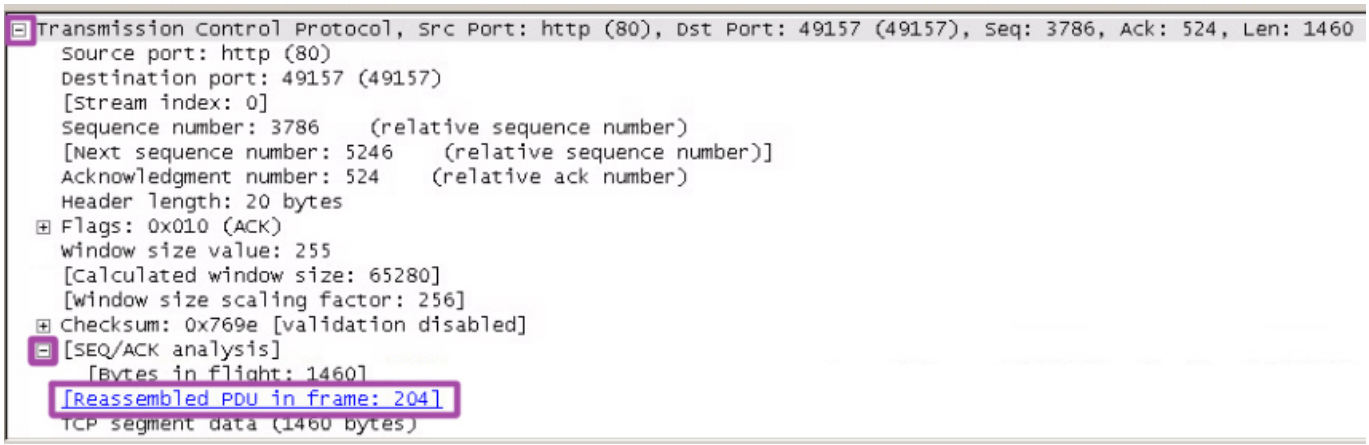

### SEQ/ACK ANALYSIS

23. **Scroll** through the list and **select** packet number 16.




### PACKET NUMBER 16

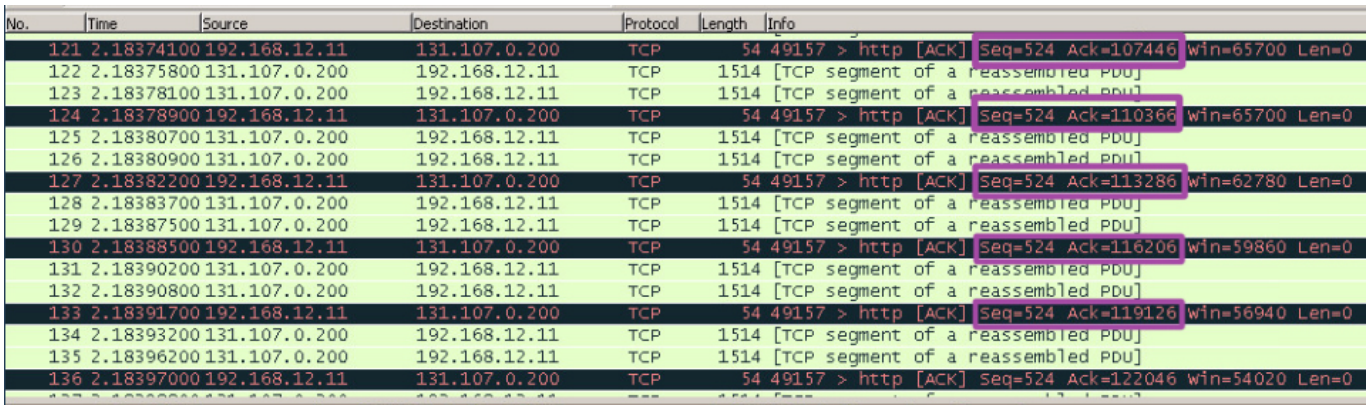
24. Using the procedures from above, **expand** the Transmission Control Protocol segment and **look** at the options that are set. **Expand** the + next to the line SEQ/ACK analysis. **Notice** the message now states Reassembled PDU in frame: 204. This means the picture was completely transmitted and reassembled in packet number 204.




### REASSEMBLED PDU IN FRAME: 204

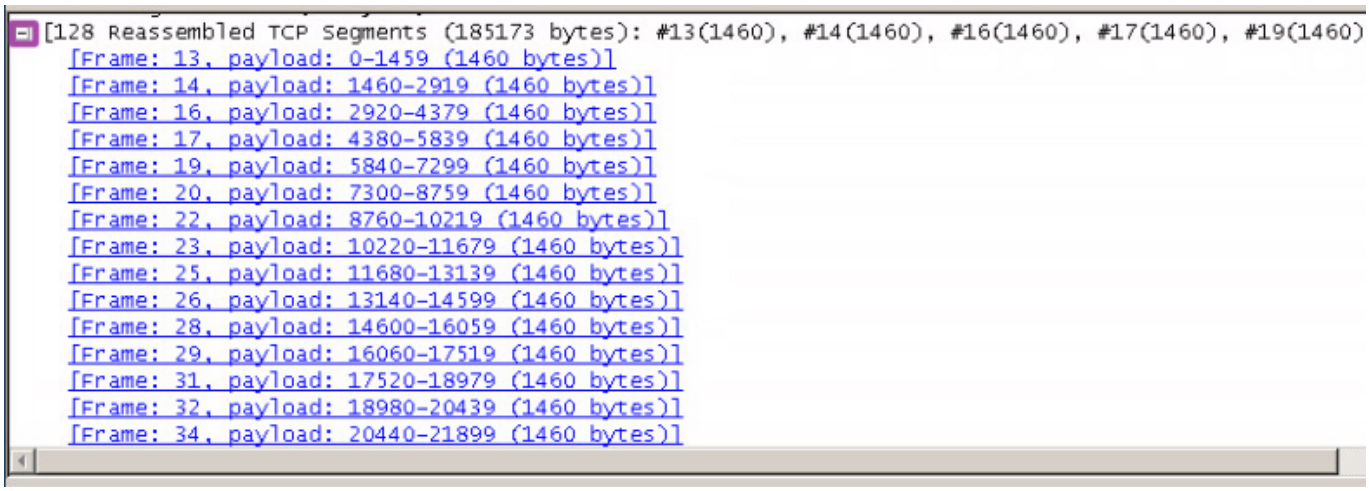
25. **Scroll** slowly down through the top capture window, noticing some of the packets that are

highlighted in black. **Look** closely at the Info column. **Notice** that each of these segments is an acknowledgment to a previous group of segments. **Remember** that the Ack number is always the next segment expected in the sequence. **Notice** that even though the Ack (acknowledgment) number continues to increase, the Seq (sequence) number does not. This is because the client computer has not sent any additional segments; it is only receiving segments from the web server.




ACK AND SEQ NUMBERS

26. **Scroll** through the top capture window and **select** packet number 204 by clicking on it. **Notice** that the protocol is once again HTTP. In the middle pane of the capture window, **expand** the + next to [128 Reassembled TCP Segments. This shows all of the packets in the capture that it took to transmit the picture from the web server to the client.




128 REASSEMBLED TCP SEGMENTS PACKETS

# CONCLUSION:

There are two protocols in the TCP/IP suite that reside at the transport layer of the OSI model – TCP and UDP. The TCP protocol is the transport layer protocol used by the HTTP protocol for reliable data transfer. TCP uses a three-way handshake to initiate a conversation and then sequence and acknowledgment numbers to keep segments in the correct order during transmission. Port numbers are used to differentiate conversations.

## DISCUSSION QUESTIONS:

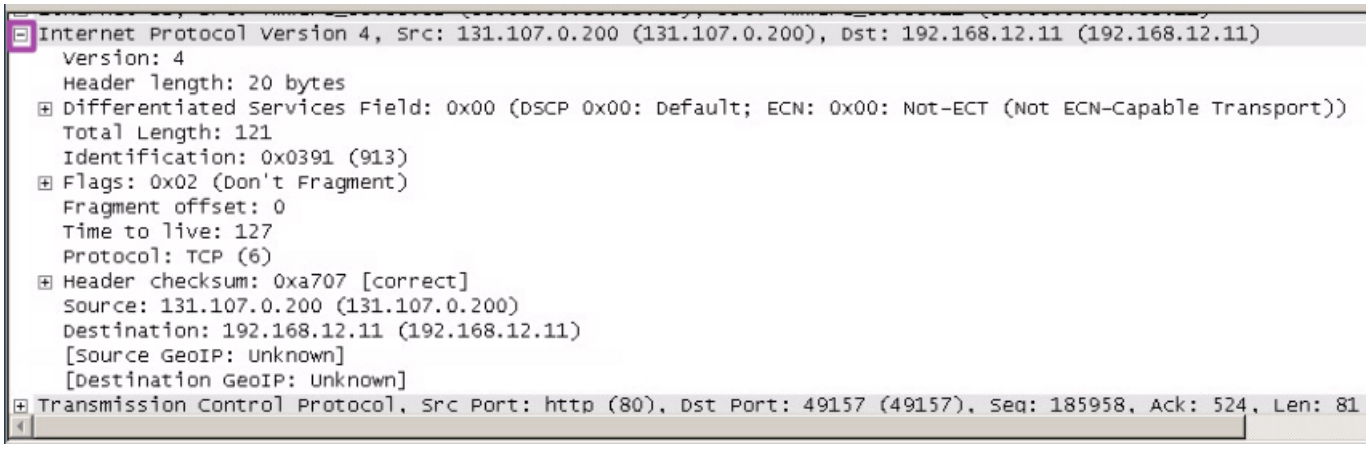
1. In Segment Protocol Data Unit section, Step 2a, what is the source port for the conversation?
2. In Segment Protocol Data Unit section, Step 2b, what is the well-known port number for the HTTP protocol?
3. Identify which flags are set in each segment of the three-way handshake. There are three segments.
4. The port number 49157 is known as this type of port because it is randomly generated when the conversation is initiated.
5. The port number 80 is known as this type of port because it is assigned to the HTTP protocol by IANA.

# Reviewing the Network Layer

The network layer of the OSI model is responsible for logical addressing. These addresses are used by routers to move packets between networks. The major protocol of the TCP/IP suite that resides at this layer is the Internet Protocol, or IP. There are currently two versions of IP – 4 and 6. IP version 4 addresses are 32-bits in length and are represented in a dotted decimal notation. An example of an IPv4 address is 192.168.12.11. IP version 6 addresses are 128 bits in length and are represented in eight groups of four hexadecimal digits each. An example of an IPv6 address is 2001:0db8:85a3:0042:1000:8a2e:0370:7334. IPv6 is quickly becoming the new norm as the IPv4 address space has been exhausted. The PDU associated with the network layer is the packet.

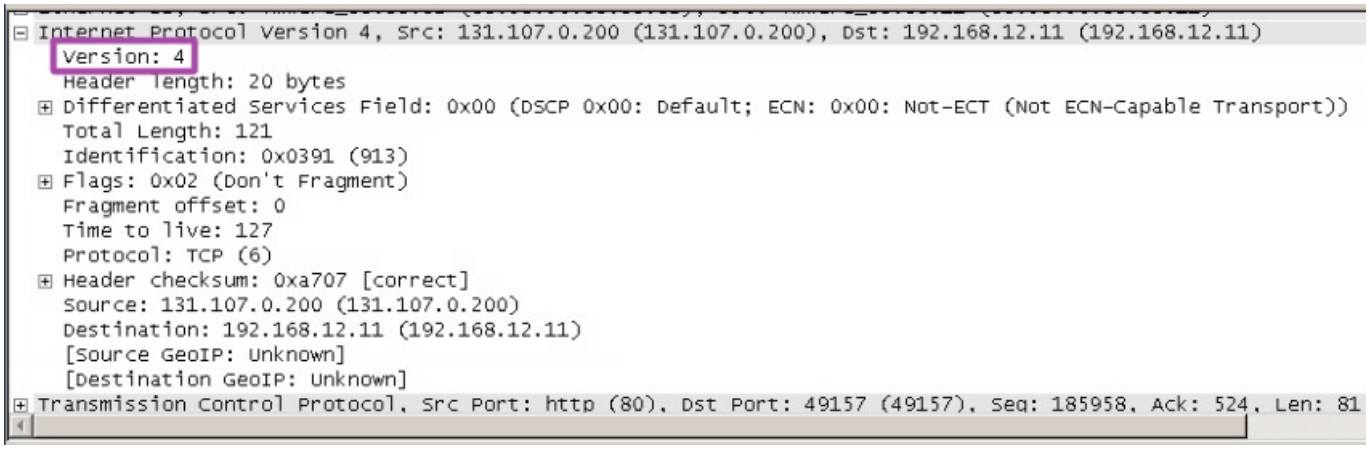
## The Packet Protocol Data Unit

1. With packet number 204 still selected, **expand** the + next to Internet Protocol Version 4. You are now viewing the IP header that has encapsulated the TCP segment. Notice the various parts that make up the IP header.



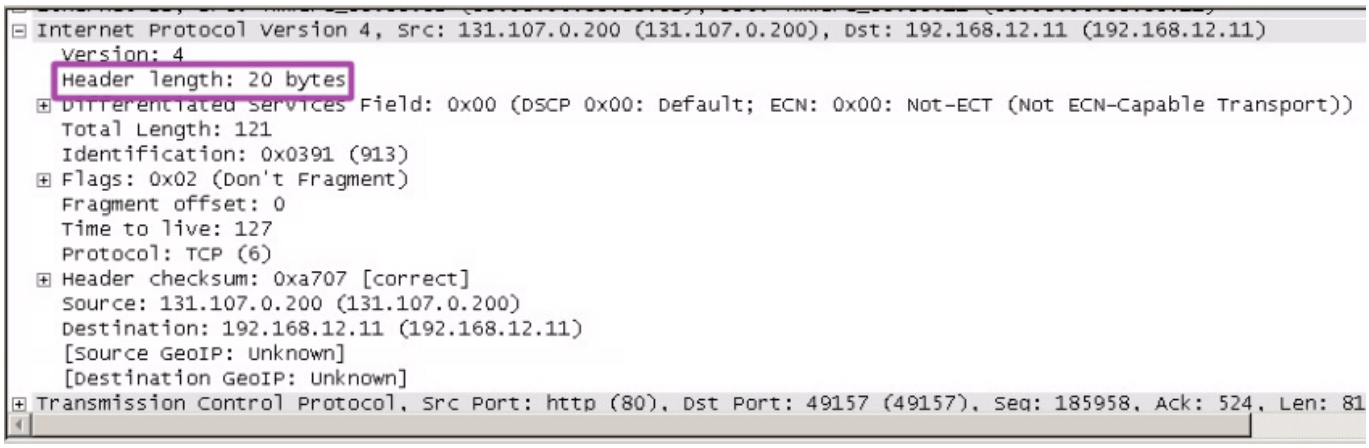

**IPV4 HEADER**

2. **Notice** the line Version. This describes the IP version in use for this packet. This line has only two options: 4 or 6. In this example, we are using version 4.



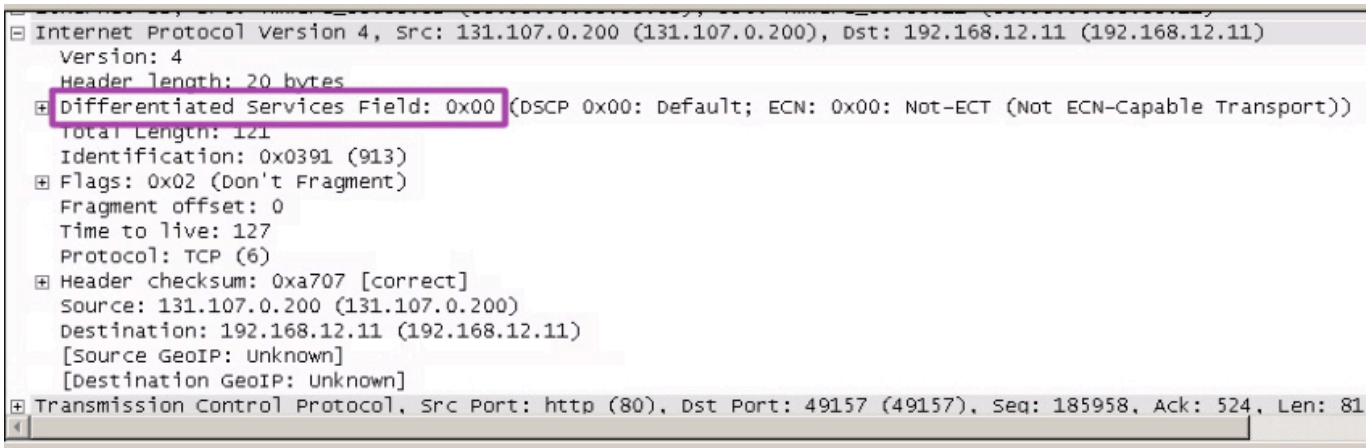

### IP VERSION

3. **Notice** the line Header length. This describes the length of the IP header only. In IPv4, this is usually 20 bytes as it is in this example.



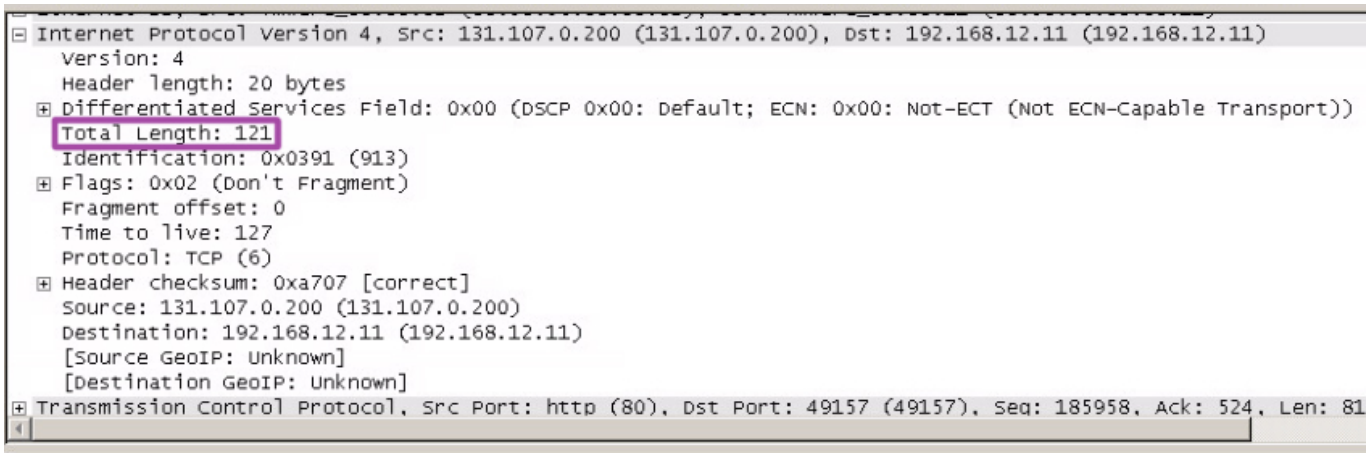

### HEADER LENGTH

4. The Differentiated Services Field can be used to specify certain Quality of Service parameters for a packet. In this example, this field is not used and set to 0x00.



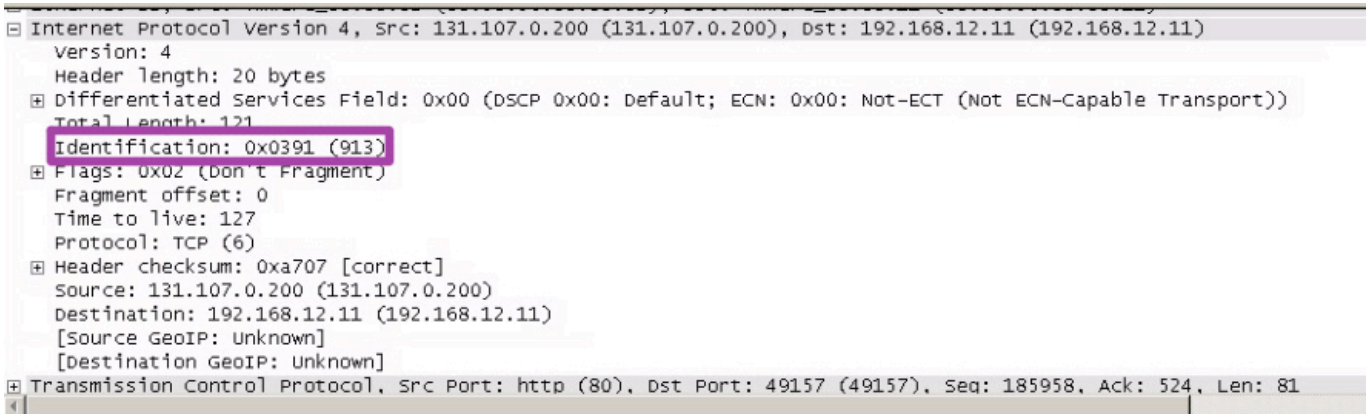

### DIFFERENTIATED SERVICES FIELD

5. **Notice** the line Total Length. This describes the length of the IP header plus the length of the segment passed down from the transport layer (in bytes). In this example, the total length of this packet is 121 bytes.



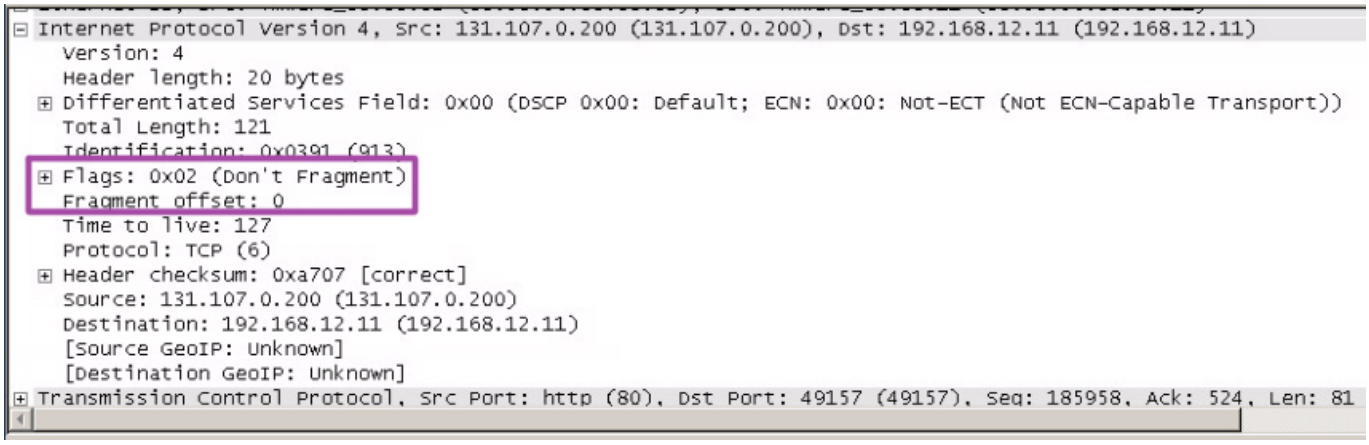

### IP TOTAL LENGTH

6. The Identification line is a 16-bit number used to uniquely identify the IP packet within the conversation. In this example, the Identification number is 0x0391. The "0x" means that the number being represented is actually in the hexadecimal format. The number in the parenthesis to the right is that same hexadecimal number converted to decimal format.




### IDENTIFICATION LINE

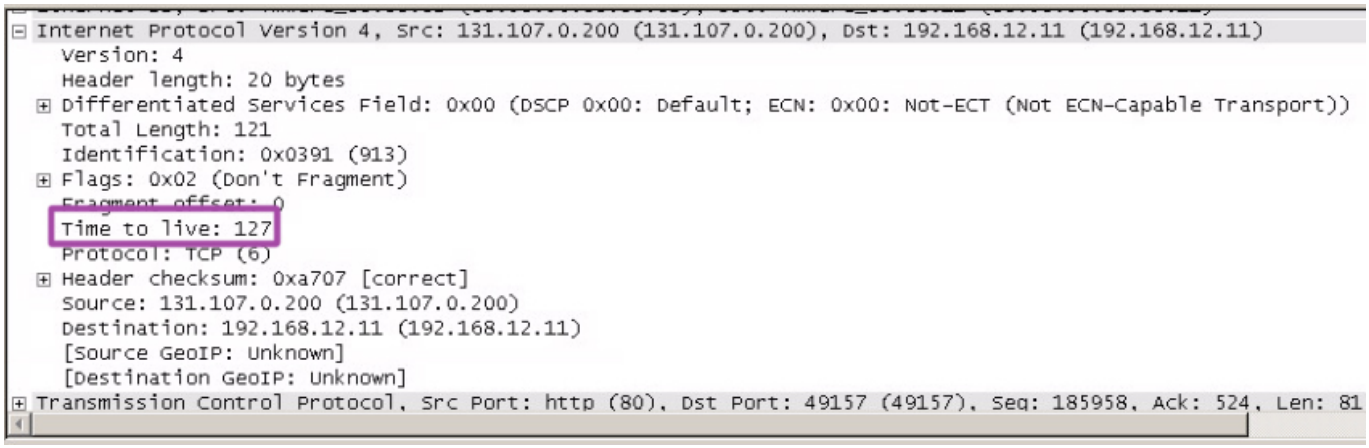
7. The Flags and Fragment offset lines go together. These fields control whether a router can fragment an IP packet and indicate the parts of the packet to the receiver. In this example, the Don't Fragment flag is set and, as such, the Fragment offset is set to 0.




### FLAGS AND FRAGMENT OFFSET LINES

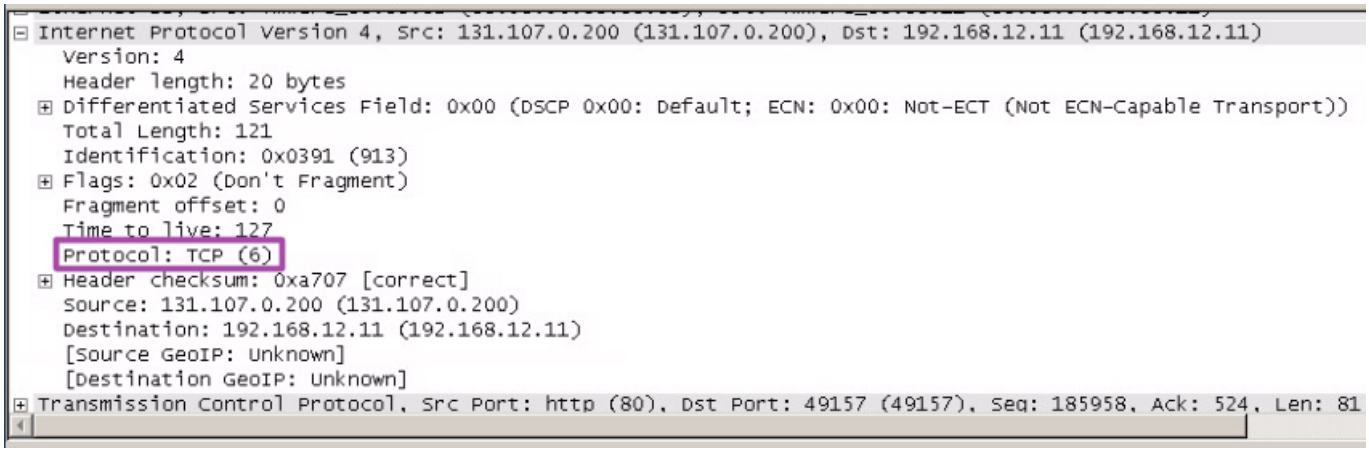
8. **Notice** the line Time to live. This number represents the number of hops (routers) that the packet

can go through on its way to the destination. Each router along the way will decrease this number by one. If this number ever reaches “0” (typically due to a routing loop) the packet will be discarded. In this example, the TTL is set to 127.



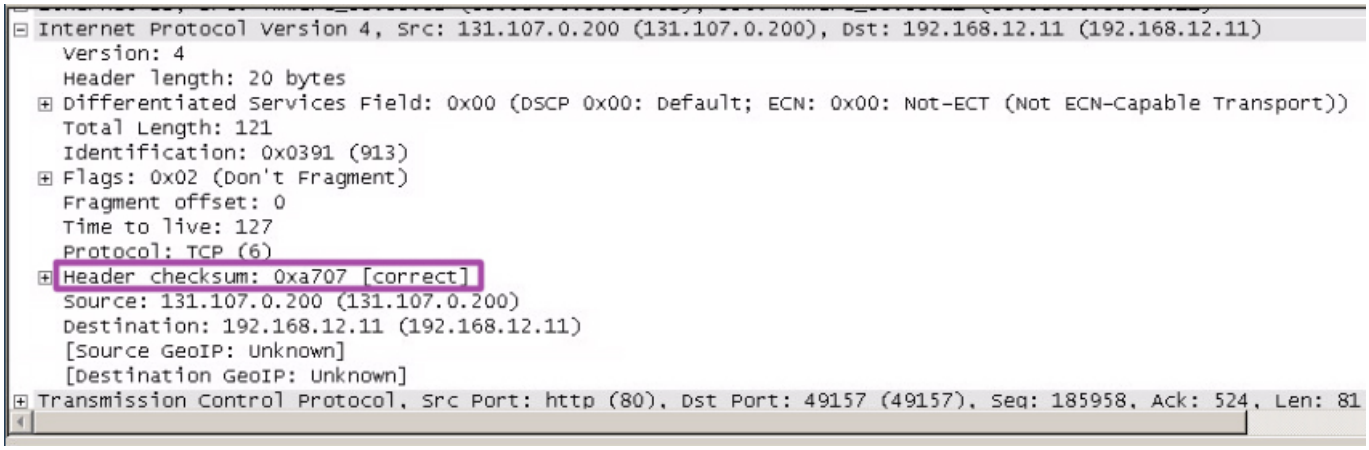

**TIME TO LIVE**

9. **Notice** the line Protocol. This field is used to indicate the transport layer protocol being carried by the packet. Each protocol is identified by a number. In this example, the protocol in use is number 6, or TCP. Another example is the UDP protocol represented by the number 17.



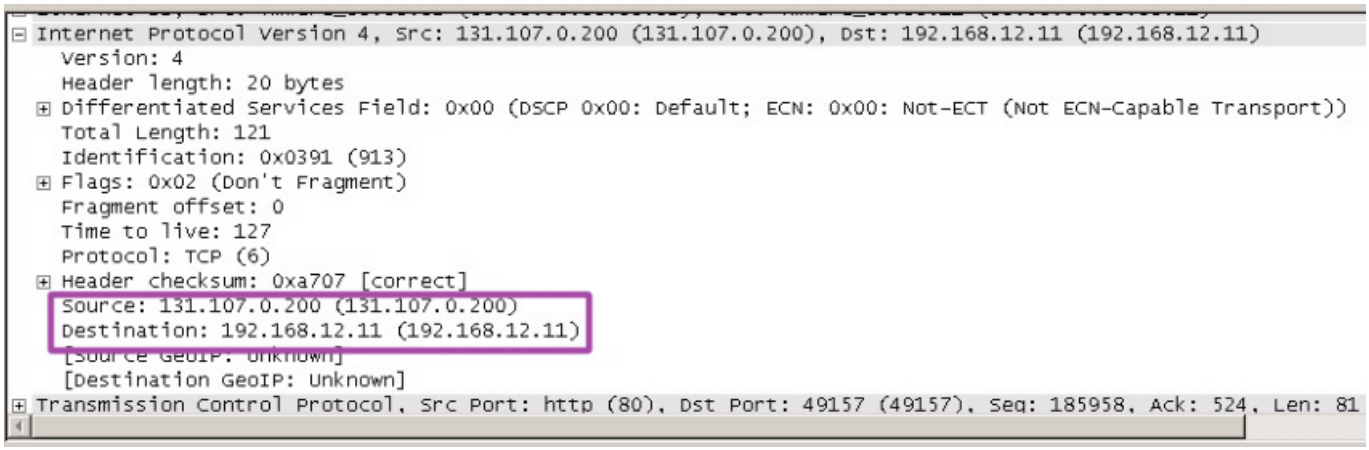

**PROTOCOL**

10. The Header checksum is used to verify that the header has not become corrupted or modified in transfer. If the checksum is correct, the packet is accepted. If the checksum is incorrect, the packet is discarded. IP is a connectionless protocol meaning that if it discards a packet, it does not ask for the packet to be retransmitted. In this example, the checksum was calculated correctly.



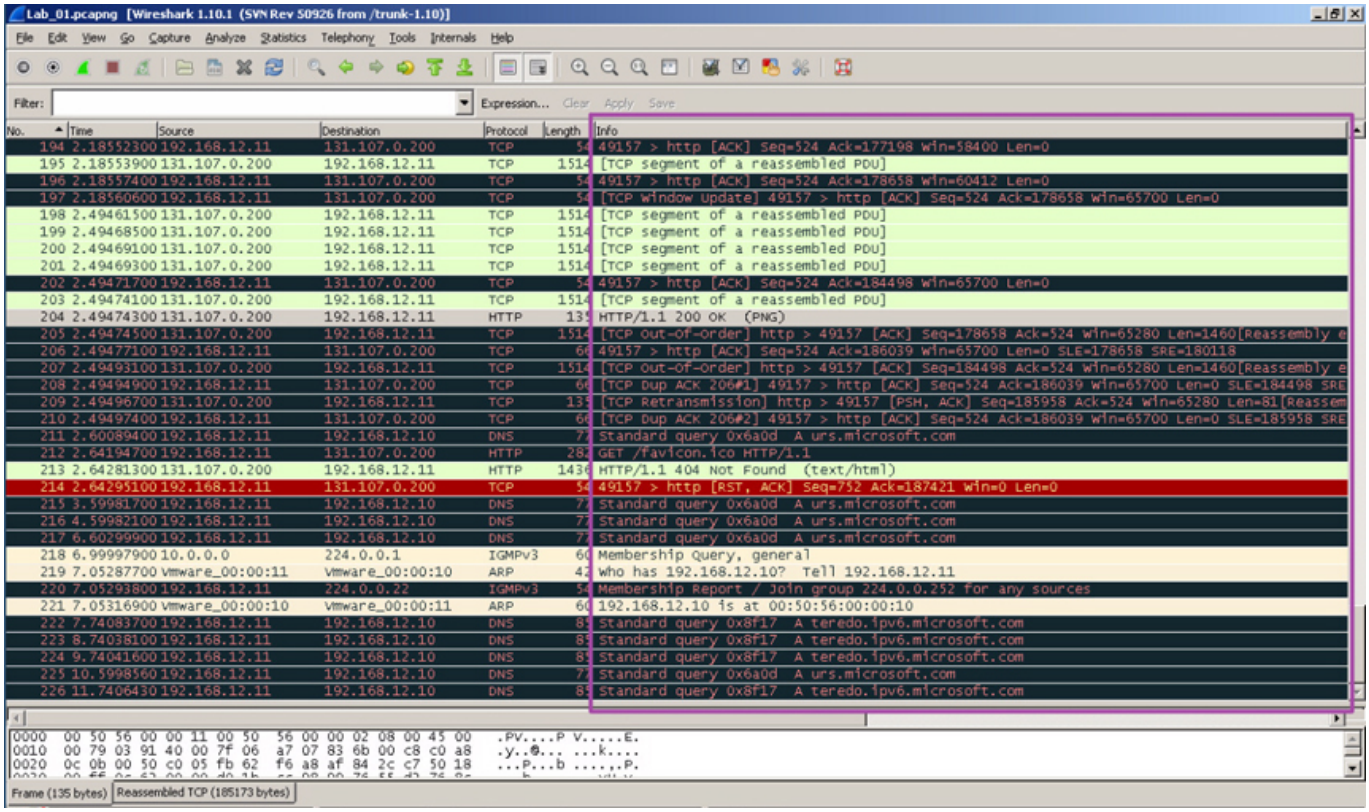

### HEADER CHECKSUM

11. **Notice** the lines Source and Destination. These represent the logical IP addresses of the client and the server. In this example, the Source IP address is the web server (131.107.0.200) and the Destination IP address is the client requesting the web page (192.168.12.11).




### SOURCE AND DESTINATION IP ADDRESSES

12. **Take** a moment to view the packet information within other captured frames. **Notice** that each of them has the same basic structure.




**OTHER CAPTURED FRAMES**

## CONCLUSION:

The network layer of the OSI model is responsible for logical addressing. IP is the major protocol of the TCP/IP suite that resides at the network layer. Routers use IP addresses to forward packets to their destination network.

## DISCUSSION QUESTIONS:

1. What is the typical header length for the IP protocol?
2. What device decreases the TTL value whenever a packet traverses it?
3. What is the PDU associated with the network layer of the OSI model?

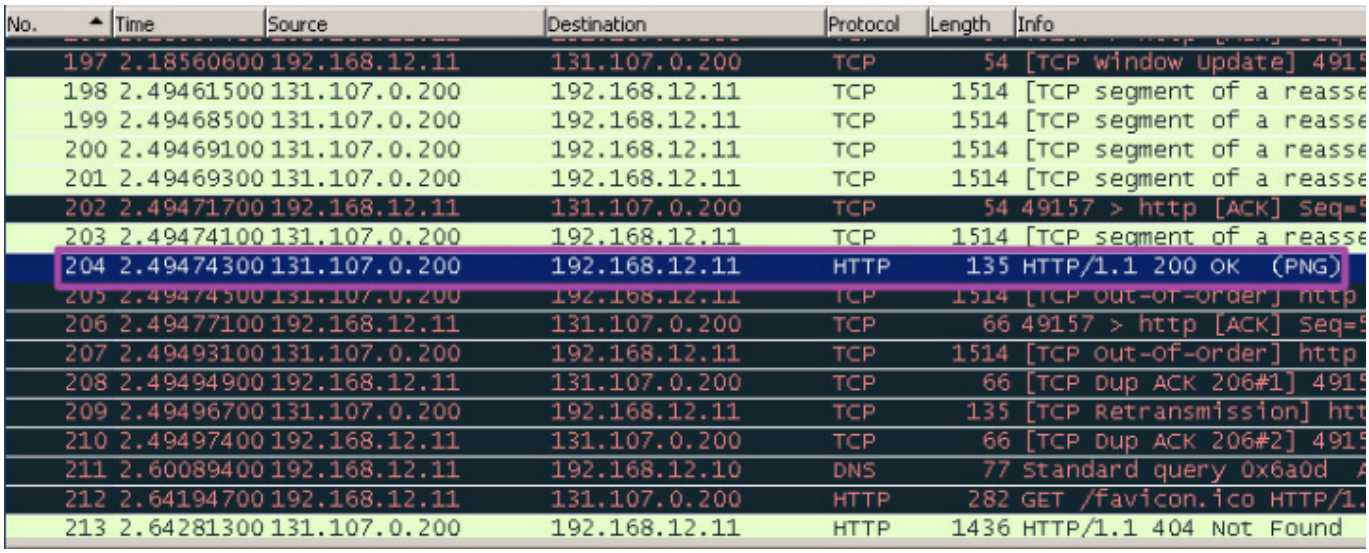
# Reviewing the Data Link Layer

The data link layer of the OSI model is responsible for physical addressing. These addresses are used by devices such as switches to move frames between nodes on the same network. One of the most common protocols that reside at this layer of the OSI model is Ethernet. Ethernet uses Media Access Control, or MAC, addresses burned into the ROM of network cards (NIC) to address its frames. MAC addresses are 48 bits, or 6 bytes, in length and are unique to every NIC. The PDU associated with the data link layer of the OSI model is the frame.

### Frame Protocol Data Unit

1. Continuing from the previous task, **select** packet number 204 in the top capture window by clicking

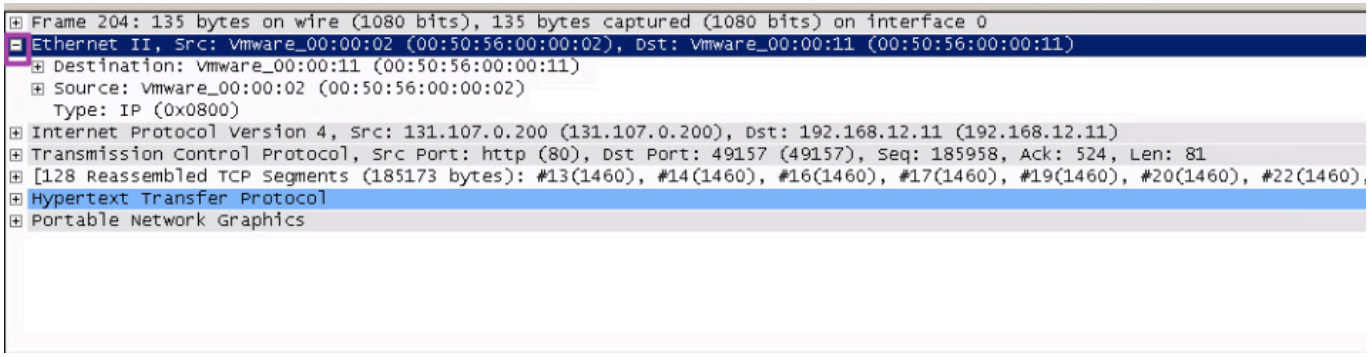
on it.



| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 197 | 2.18560600 | 192.168.12.11 | 131.107.0.200 | TCP | 54 | [TCP Window Update] 491 |
| 198 | 2.49461500 | 131.107.0.200 | 192.168.12.11 | TCP | 1514 | [TCP segment of a reasse |
| 199 | 2.49468500 | 131.107.0.200 | 192.168.12.11 | TCP | 1514 | [TCP segment of a reasse |
| 200 | 2.49469100 | 131.107.0.200 | 192.168.12.11 | TCP | 1514 | [TCP segment of a reasse |
| 201 | 2.49469300 | 131.107.0.200 | 192.168.12.11 | TCP | 1514 | [TCP segment of a reasse |
| 202 | 2.49471700 | 192.168.12.11 | 131.107.0.200 | TCP | 54 | 49157 > http [ACK] Seq=5 |
| 203 | 2.49474100 | 131.107.0.200 | 192.168.12.11 | TCP | 1514 | [TCP segment of a reasse |
| 204 | 2.49474300 | 131.107.0.200 | 192.168.12.11 | HTTP | 135 | HTTP/1.1 200 OK (PNG) |
| 205 | 2.49474500 | 131.107.0.200 | 192.168.12.11 | TCP | 1514 | [TCP Out-Of-Order] http |
| 206 | 2.49477100 | 192.168.12.11 | 131.107.0.200 | TCP | 66 | 49157 > http [ACK] Seq=5 |
| 207 | 2.49493100 | 131.107.0.200 | 192.168.12.11 | TCP | 1514 | [TCP Out-of-Order] http |
| 208 | 2.49494900 | 192.168.12.11 | 131.107.0.200 | TCP | 66 | [TCP Dup ACK 206#1] 491 |
| 209 | 2.49496700 | 131.107.0.200 | 192.168.12.11 | TCP | 135 | [TCP Retransmission] htt |
| 210 | 2.49497400 | 192.168.12.11 | 131.107.0.200 | TCP | 66 | [TCP Dup ACK 206#2] 491 |
| 211 | 2.60089400 | 192.168.12.11 | 192.168.12.10 | DNS | 77 | Standard query 0x6a0d A |
| 212 | 2.64194700 | 192.168.12.11 | 131.107.0.200 | HTTP | 282 | GET /favicon.ico HTTP/1. |
| 213 | 2.64281300 | 131.107.0.200 | 192.168.12.11 | HTTP | 1436 | HTTP/1.1 404 Not Found |

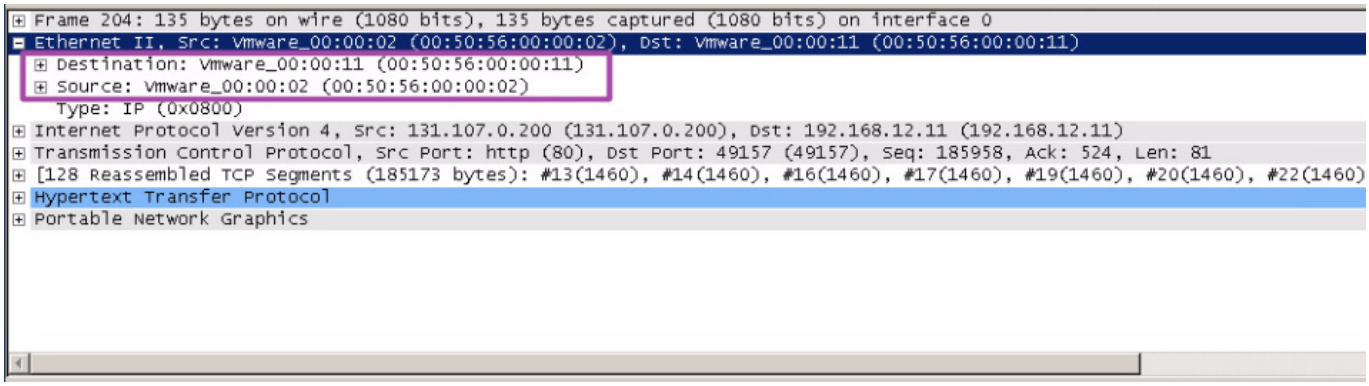
### PACKET NUMBER 204

2. In the middle pane of the capture window, **expand** the + next to Ethernet II. This line describes the layer 2 protocol that is used to encapsulate the IP packet and prepare the information to be transmitted over the physical media.




### ETHERNET II

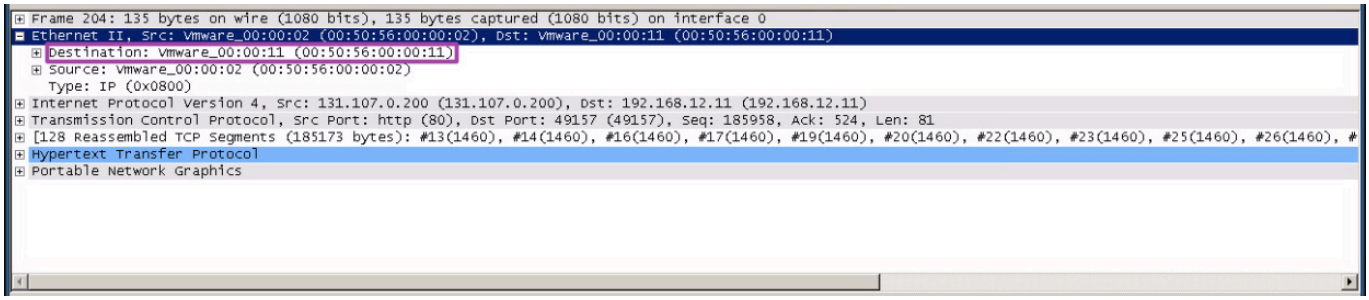
3. **Notice** the lines Destination and Source. Ethernet uses the physical MAC addresses burned into the NIC as the addresses at this layer.




### DESTINATION AND SOURCE LINES

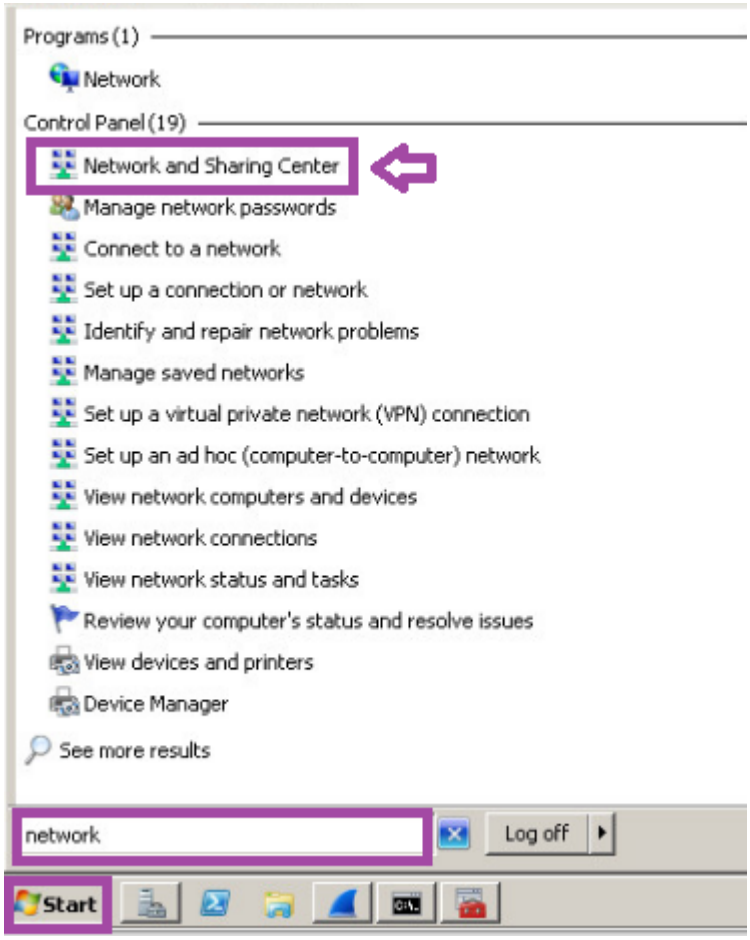
4. The Destination address in this example is the MAC address of the client machine (look at the one in parenthesis). The first 3 bytes of the MAC address represent the Organizationally Unique Identifier

(OUI). This is a number assigned by IEEE to identify the vendor of the NIC. Wireshark automatically substitutes the vendor's name for the first 3 bytes of the MAC address. The last 3 bytes of the MAC address is essentially a serial number assigned to the NIC by the manufacturer.



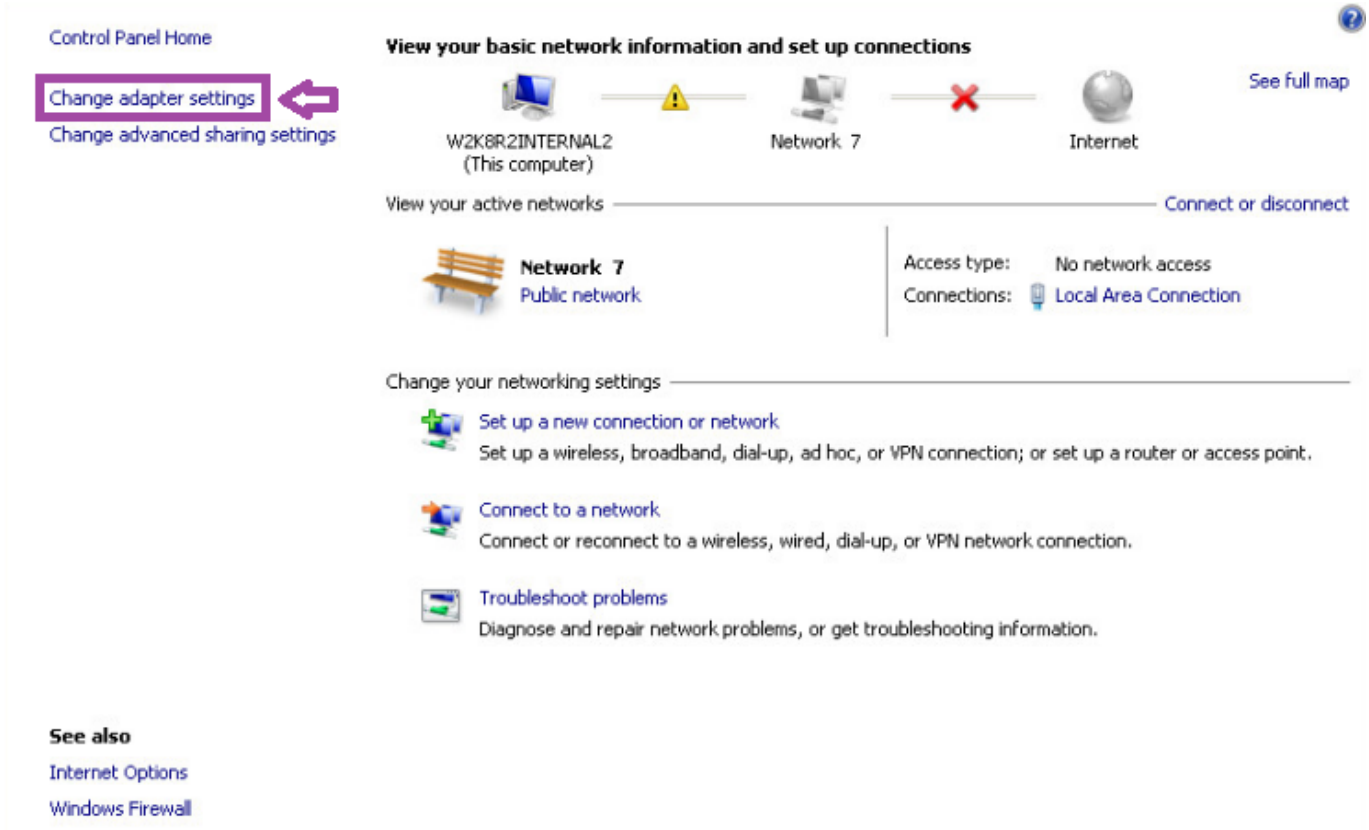

### DESTINATION ADDRESS

5. Here is an example of how it is possible to filter or change the MAC address of a client machine. MAC adress filtering is used in some networks to control which end-user devices or computers can connect to the network. **Click** on the Start button and **type** `network` in the search box. Then **click** on Network and Sharing Center in the search results.



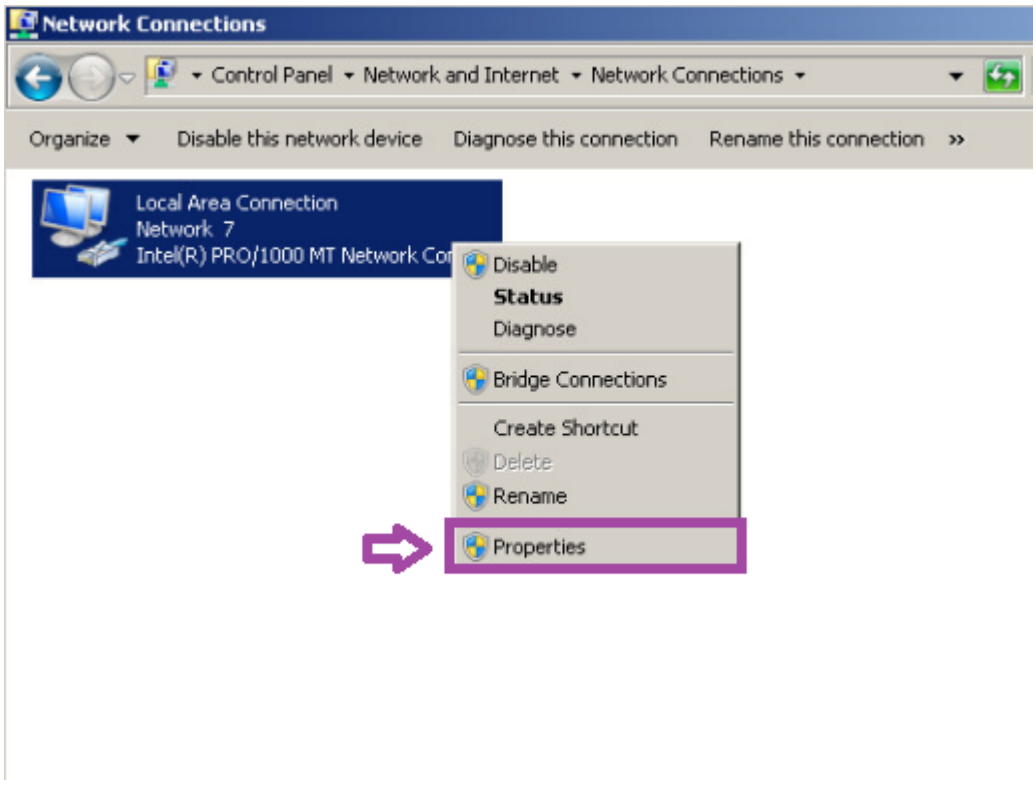

### NETWORK AND SHARING CENTER

6. Next, **click** on Change adapter settings.



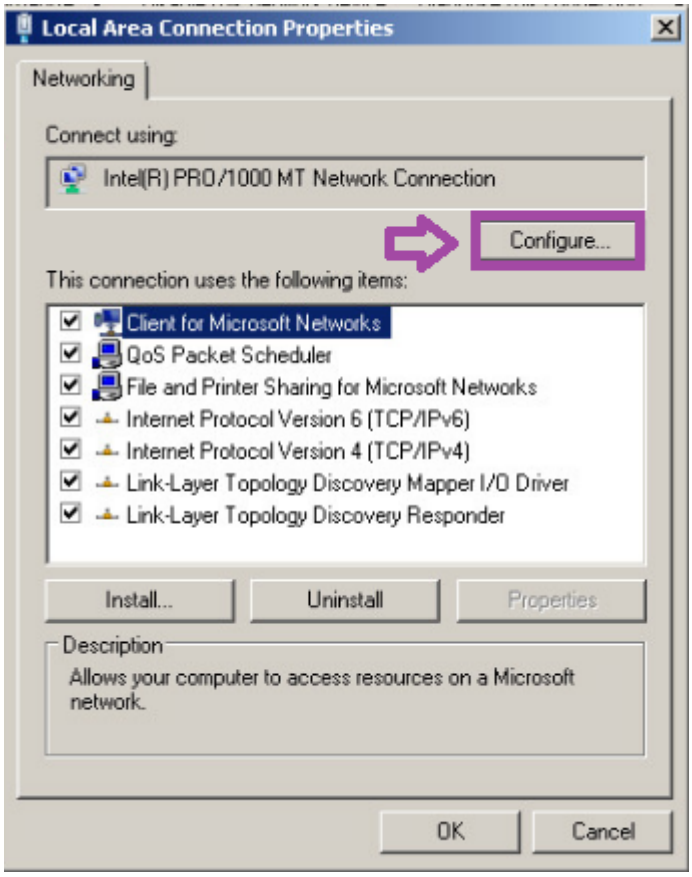

### CHANGE ADAPTER SETTINGS

7. **Right-click** on Local Area Connection, and then **select** Properties.



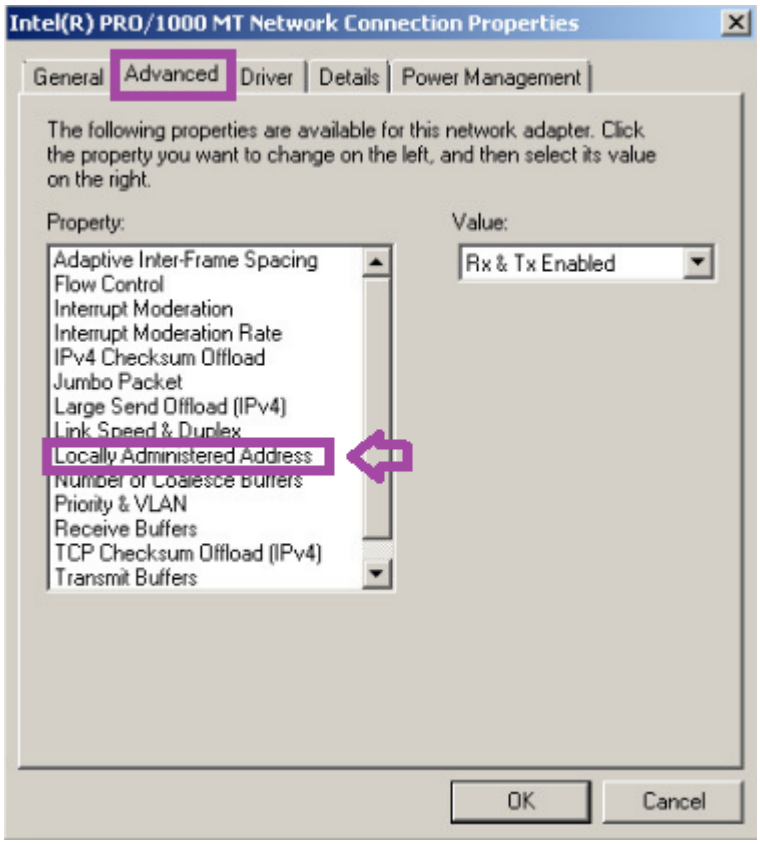

### LOCAL AREA CONNECTION

8. Next, **click** the Configure button.



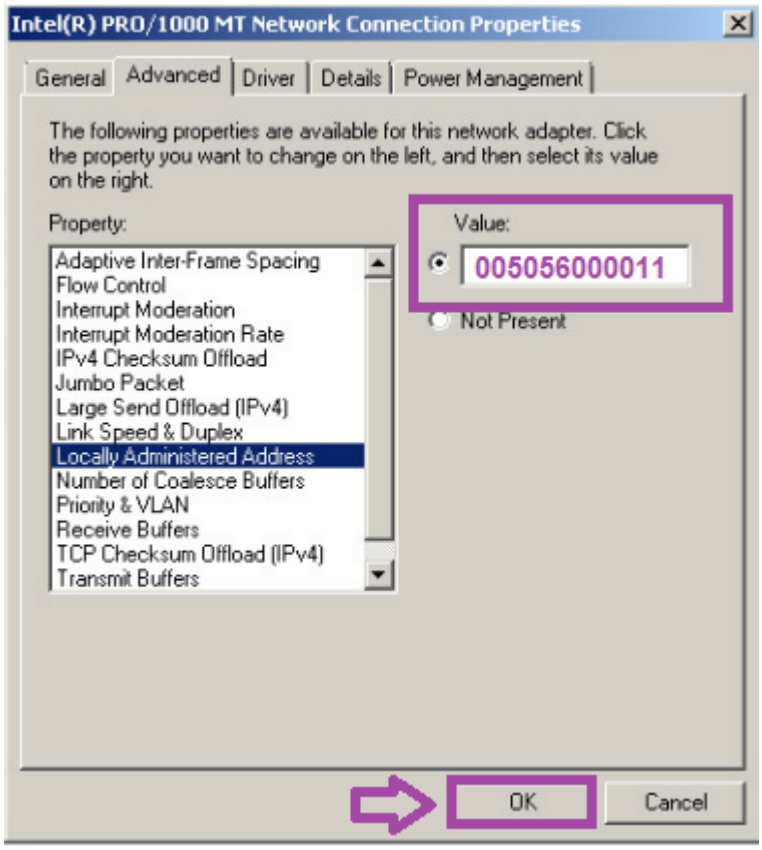

LOCAL AREA CONNECTION PROPERTIES

9. **Click** on the Advanced tab, then under Property: **select** Locally Administered Address.



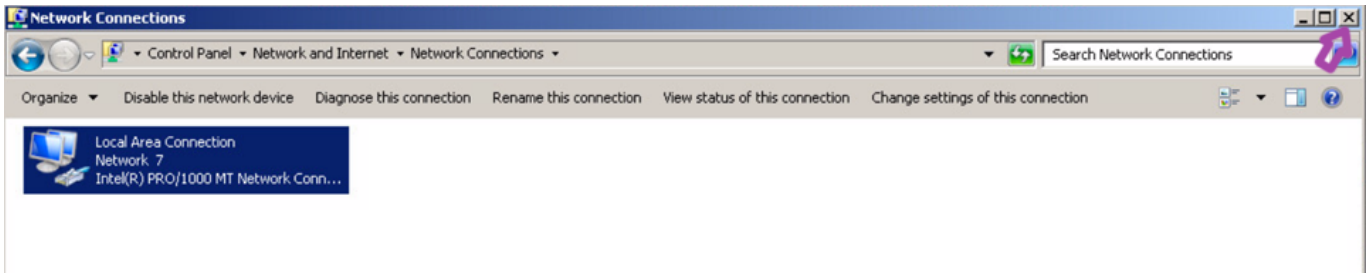

INTELL(R) PRO/1000 NETWORK CONNECTION PROPERTIES

10. **Select** the Value indicator, then **type** the MAC Address (without dashes): `005056000011` and then **click** OK.



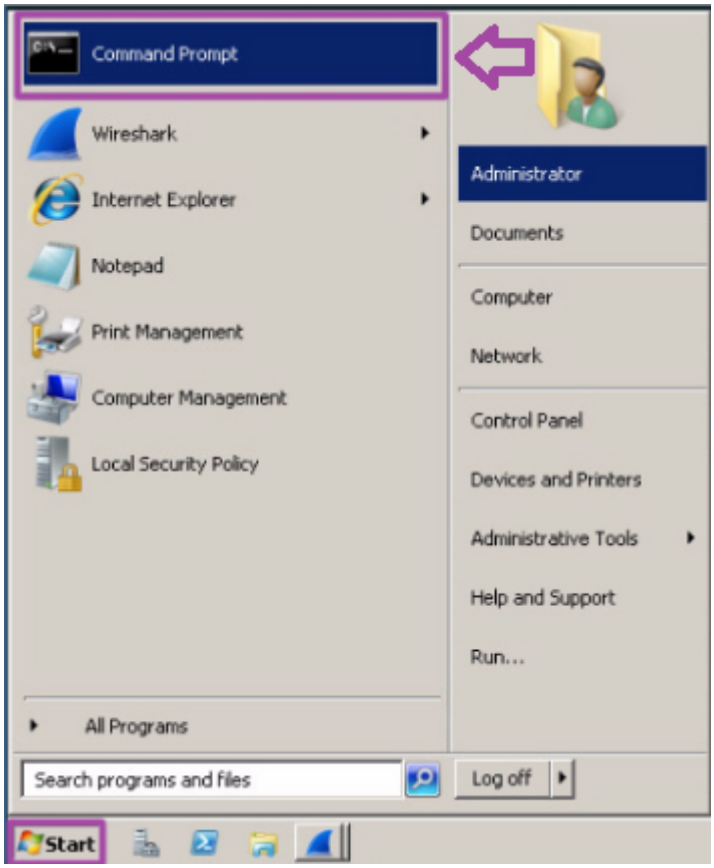

INTELL(R) PRO/1000 NETWORK CONNECTION PROPERTIES

11. **Click** the X in the upper-right corner to close Network Connections.



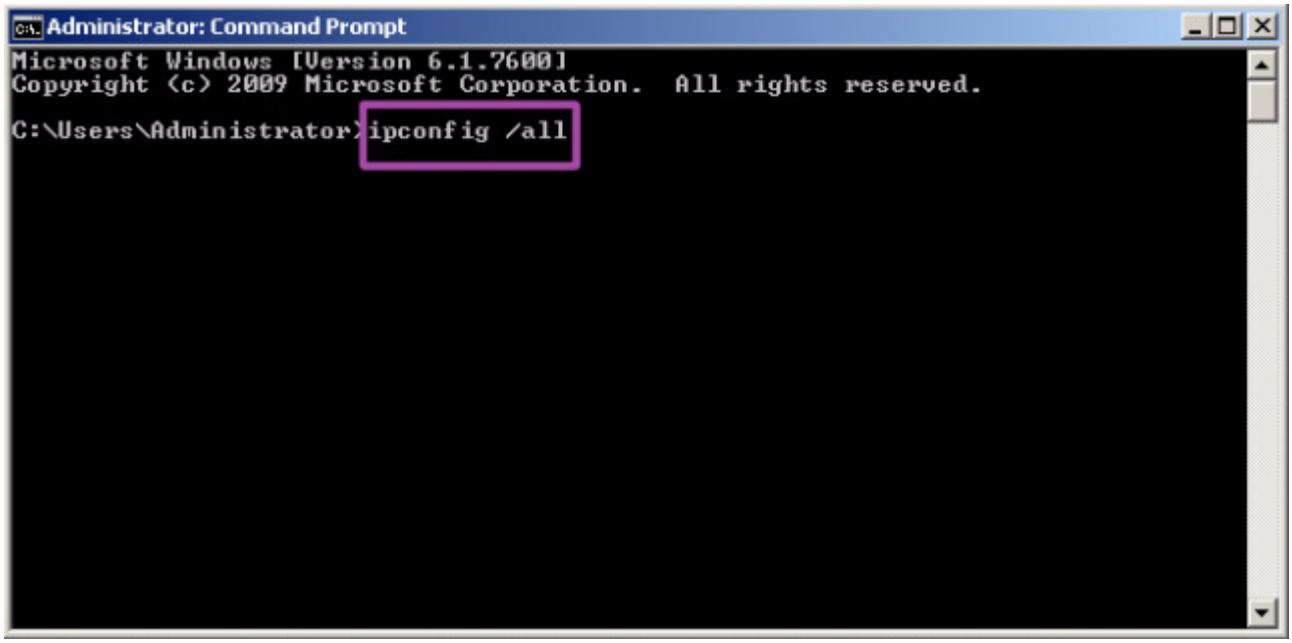

CLOSING NETWORK CONNECTIONS

12. Let's take a moment to verify the MAC address for the client machine. **Click** on the Start button, then **click** on the Command Prompt shortcut.




COMMAND PROMPT

13. In the Command Prompt window, **type** the command `ipconfig /all` and **press** Enter.




COMMAND PROMPT

14. **Scroll up** until you see the header Ethernet adapter Local Area Connection. Below this header, locate the line Physical Address. You will notice that the destination MAC address matches the Physical Address of this machine. In order for this machine to process a frame, the destination MAC address must match the Physical Address or the frame will be ignored. Note that your MAC address may vary based on your hardware.

```
C:\Windows\System32>ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : W2K8R2Internal2
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : netplus.com

Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : netplus.com
   Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Network Connection
   Physical Address. . . . . . . . . : 00-50-56-00-00-11
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::c124:4f54:a6cf:3513%11(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.12.11(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.12.1
   DHCPv6 IAID . . . . . . . . . . . : 234901590
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-18-D2-A7-38-00-50-56-9C-27-3D

   DNS Servers . . . . . . . . . . . : 192.168.12.10
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

PHYSICAL ADDRESS

15. **Type** `exit` and **press** Enter to close the Command Prompt.

```
C:\Windows\System32>exit
```

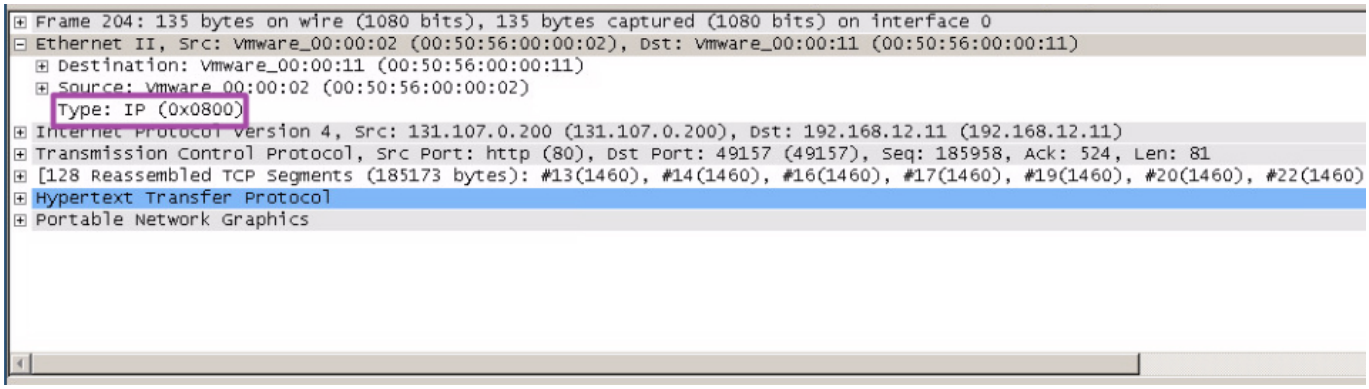
CLOSING THE COMMAND PROMPT

16. The Source address in this example is the MAC address of the default gateway for this network. The reason for this is because the data link layer information must be recreated by every router that receives the frame. Since routers make their decision on where to move packets based upon IP addresses, the data link layer information must be removed by the router so it can see the network layer information. Routers simply create new data link layer information based upon the type of network they will be sending the frame onto.

```
⊞ Frame 204: 135 bytes on wire (1080 bits), 135 bytes captured (1080 bits) on interface 0
⊟ Ethernet II, Src: Vmware_00:00:02 (00:50:56:00:00:02), Dst: Vmware_00:00:11 (00:50:56:00:00:11)
  ⊞ Destination: Vmware_00:00:11 (00:50:56:00:00:11)
  ⊞ Source: Vmware_00:00:02 (00:50:56:00:00:02)
    Type: IP (0x0800)
⊞ Internet Protocol Version 4, Src: 131.107.0.200 (131.107.0.200), Dst: 192.168.12.11 (192.168.12.11)
⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: 49157 (49157), Seq: 185958, Ack: 524, Len: 81
⊞ [128 Reassembled TCP Segments (185173 bytes): #13(1460), #14(1460), #16(1460), #17(1460), #19(1460), #20(1460)
⊞ Hypertext Transfer Protocol
⊞ Portable Network Graphics
```

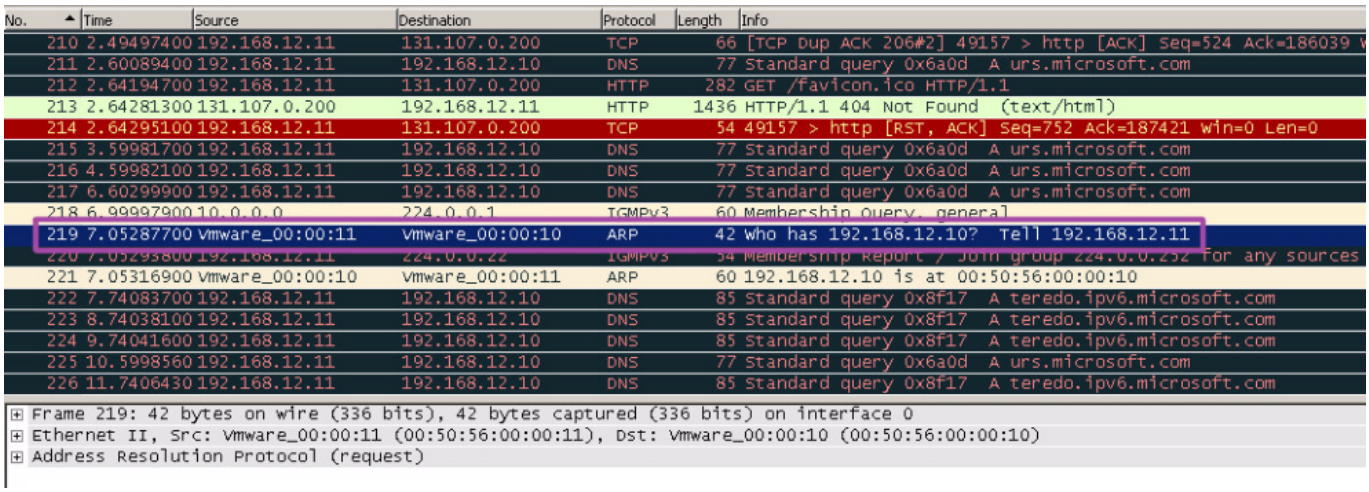
SOURCE ADDRESS

17. **Notice** the line Type. This line describes the network layer protocol that is being encapsulated by this frame. In this example, the network layer protocol being used is IP. It is represented in the frame in its hexadecimal format 0x0800.




**TYPE**

18. **Take** a moment to view the frame information within other captured frames. Notice that each of them has the same basic structure. Can you find any frames that carry a different protocol type? An example is frame 219, which carries the ARP protocol at layer 3 instead of IP.




**FRAME 219**

# CONCLUSION:

The data link layer of the OSI model is responsible for physical addressing. Ethernet switches and hosts use MAC addresses to move frames between nodes on the same network.

# DISCUSSION QUESTIONS:

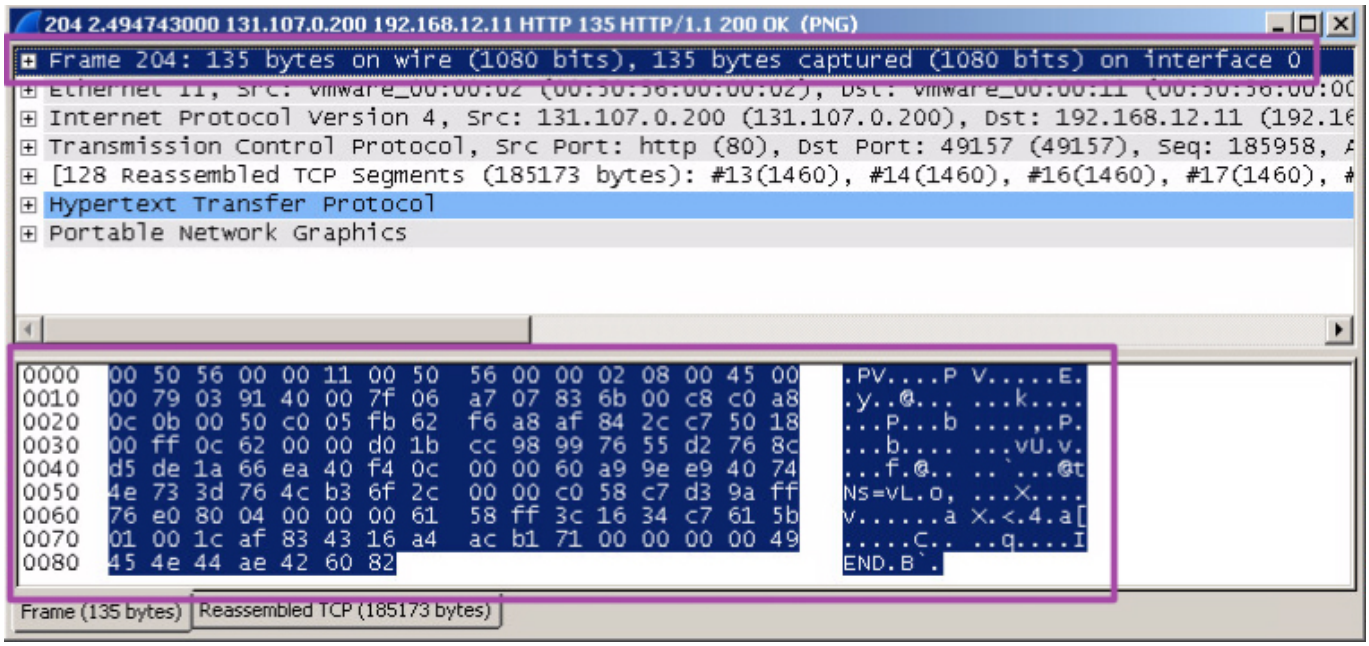
1. How long is a MAC address in bits?
2. How long is a MAC address in bytes?
3. What is the first half of a MAC address known as?
4. What command can be used on a Windows machine to view the MAC address?

# Reviewing the Physical Layer

The physical layer of the OSI model incorporates all of the tangible network components such as cables, connectors, and repeaters. The electronic signals on the media are also included in this layer. The electronic signals make the 1's and 0's that physically represent the data on the media. The PDU associated with the physical layer of the OSI model is simply bits.

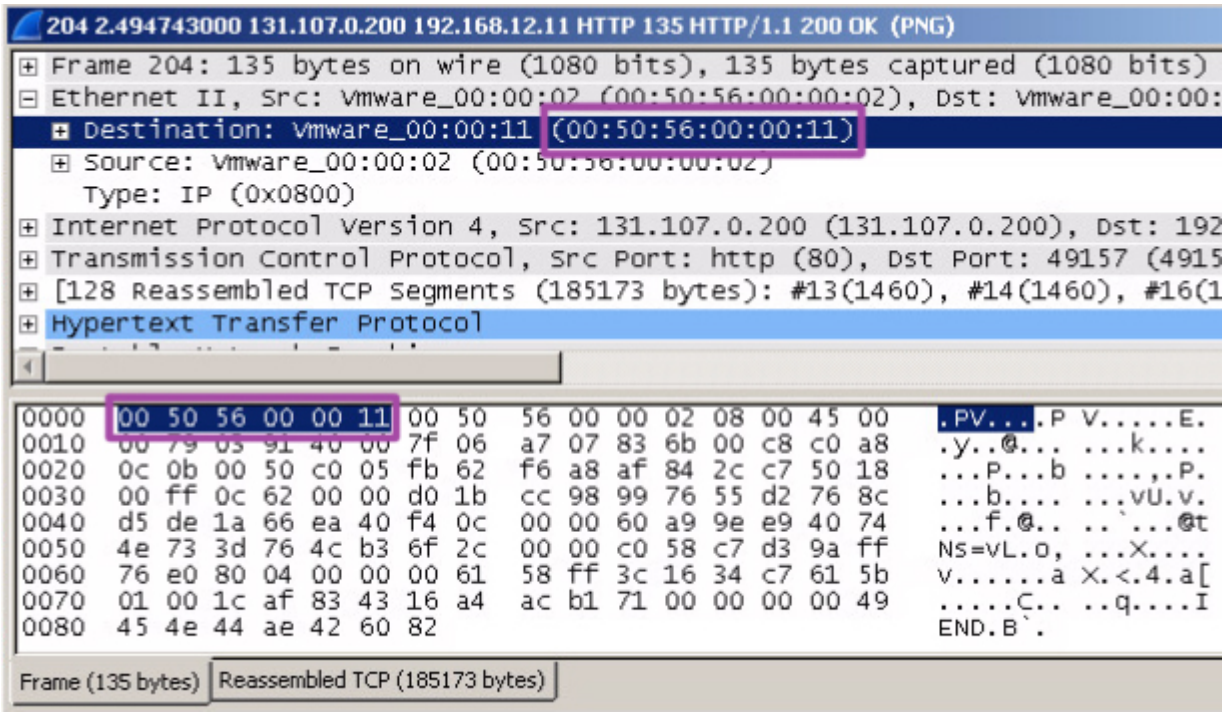
## Bit Protocol Data Unit

1. Continuing from the previous task, **select** packet number 204 in the top capture window by clicking on it (if necessary). In the very bottom window, you will notice many numbers and letters. This is the actual data represented in hexadecimal format. It is represented in this fashion simply because it is more "user-friendly" than having to read the data in its true binary fashion. One thing you may have noticed is that as you were clicking on the different sections of the packet, different pieces of the data were being highlighted. Wireshark is actually showing you where the data is located in the physical frame.



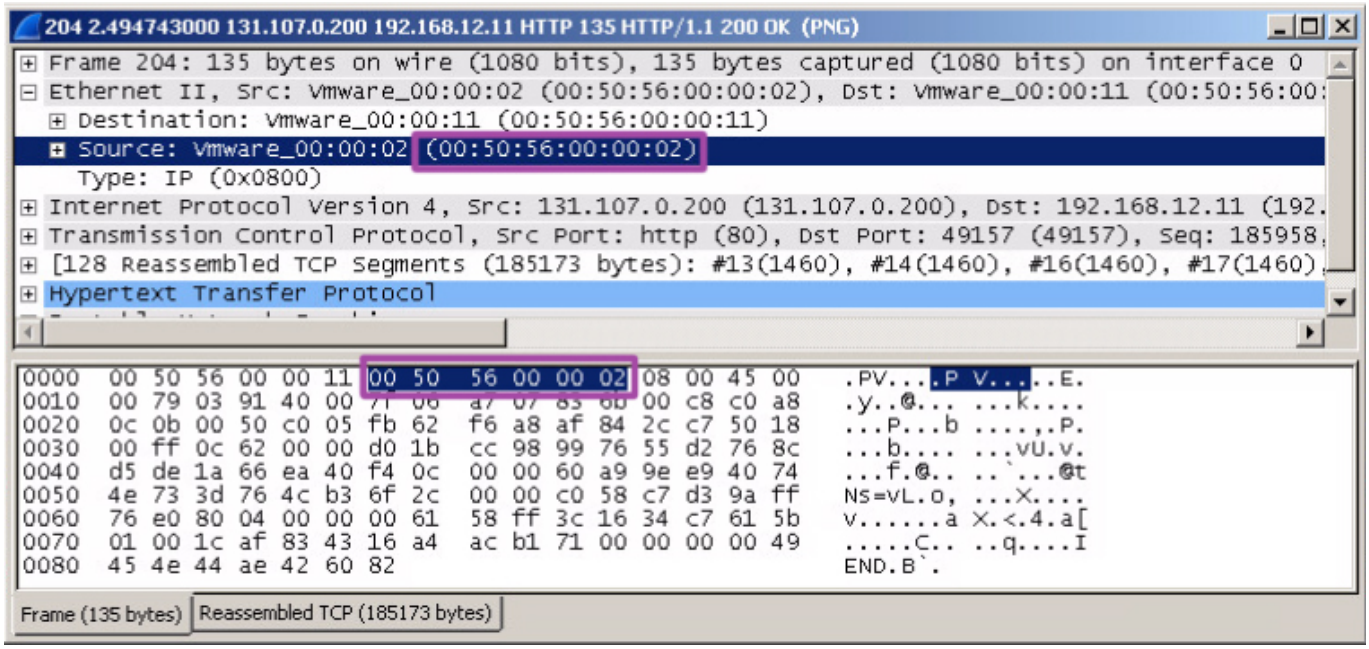

**PHYSICAL FRAME**

2. **Locate** the line Destination under the Ethernet II heading and **click** on it. Wireshark highlights the Destination MAC address of the frame within the hexadecimal data. Notice how the numbers actually match.



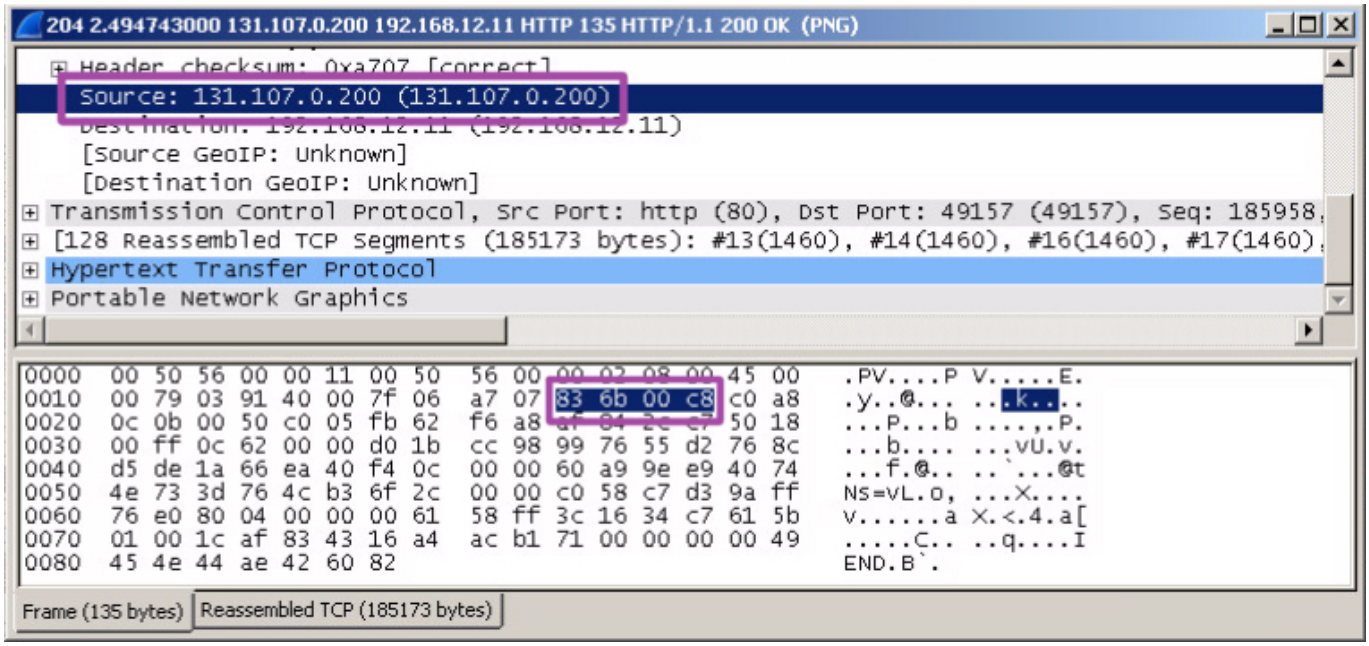

### DESTINATION MAC ADDRESS

3. **Locate** the line Source under the Ethernet II heading and **click** on it. Wireshark highlights the Source MAC address of the frame within the hexadecimal data.



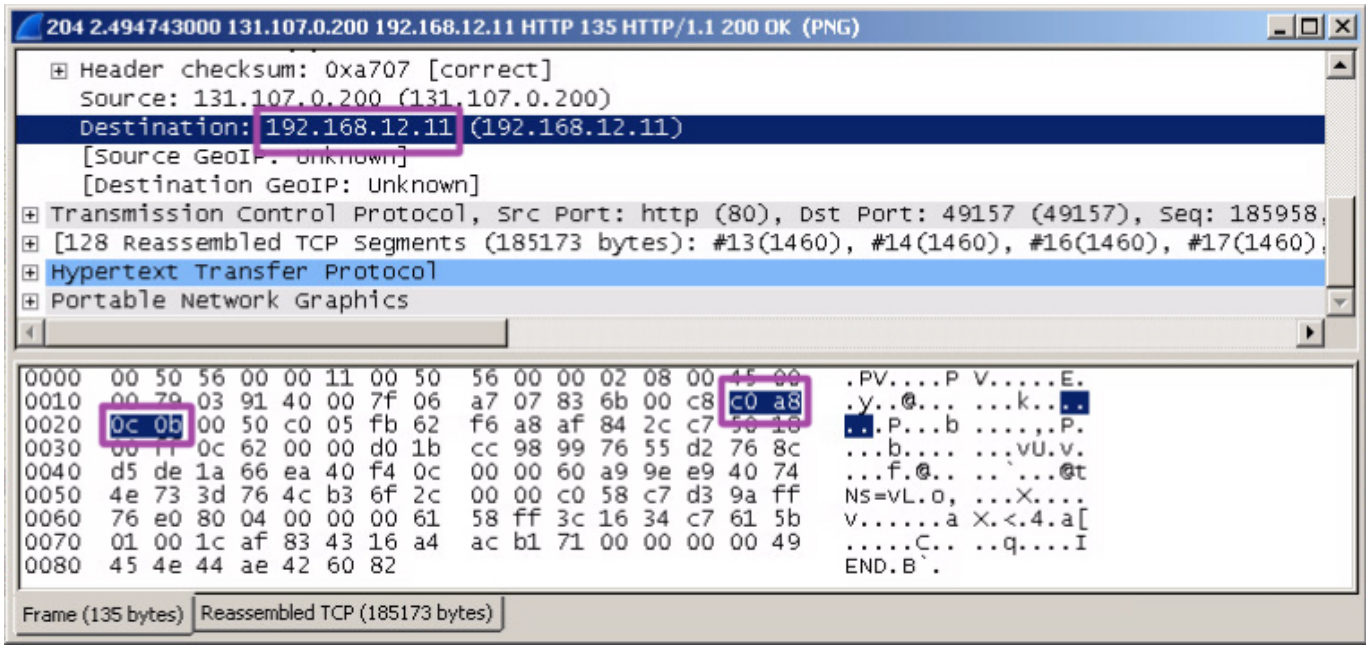

### SOURCE MAC ADDRESS

4. **Locate** the line Source under the Internet Protocol Version 4 heading and **click** on it. Wireshark highlights the Source IP address of the packet within the hexadecimal data. **Notice** that these numbers do not match exactly. This is because the IP address you are used to seeing is represented in decimal format; the raw data is represented in hexadecimal format. Each octet has been converted separately. In this example, 131.107.0.200 is represented as 83 6b 00 c8.



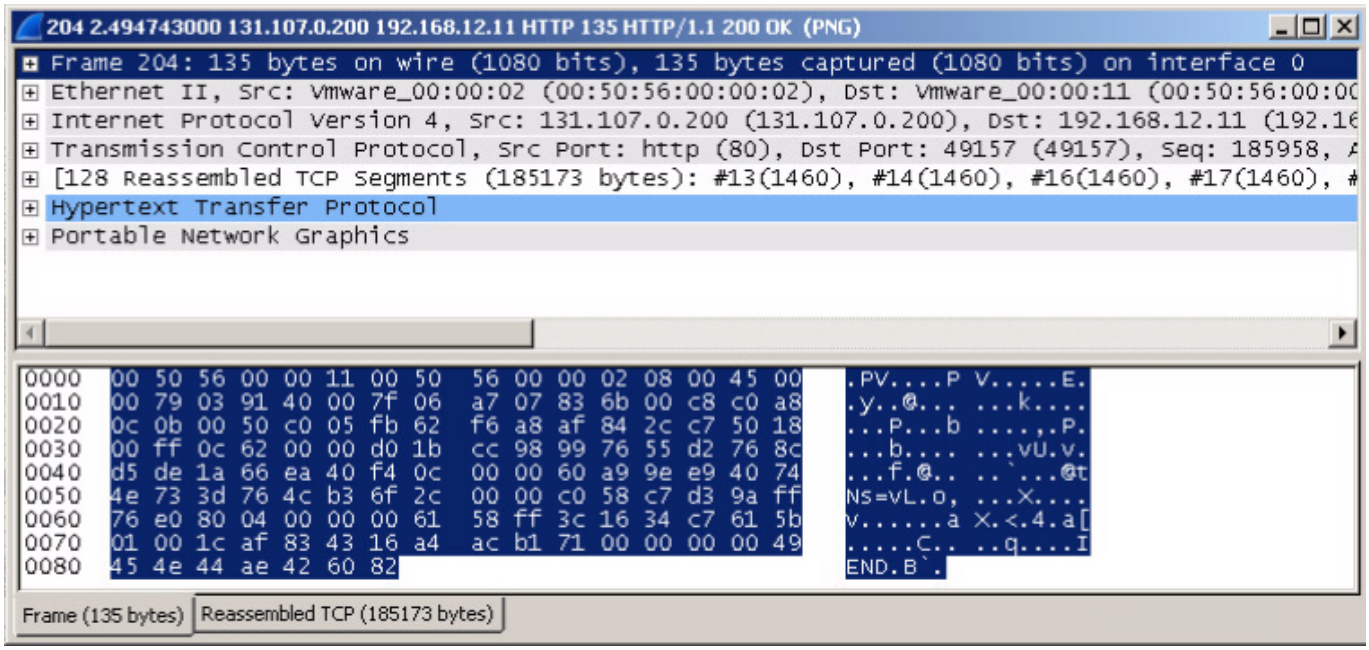

### SOURCE IP ADDRESS

5. **Locate** the line Destination under the Internet Protocol Version 4 heading. Wireshark highlights the Destination IP address of the packet within the hexadecimal data. In this example, 192.168.12.11 is represented as c0 a8 0c 0b.



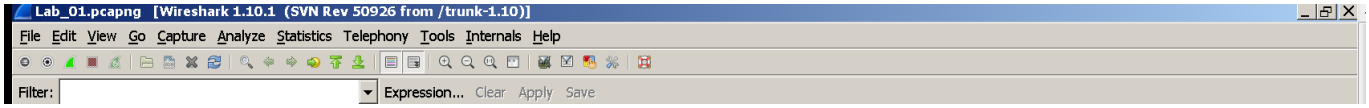

### DESTINATION IP ADDRESS

6. **Take** a moment to view some of the other information in its raw format. Keep in mind that some of the values may already be in their hexadecimal format while others may not.



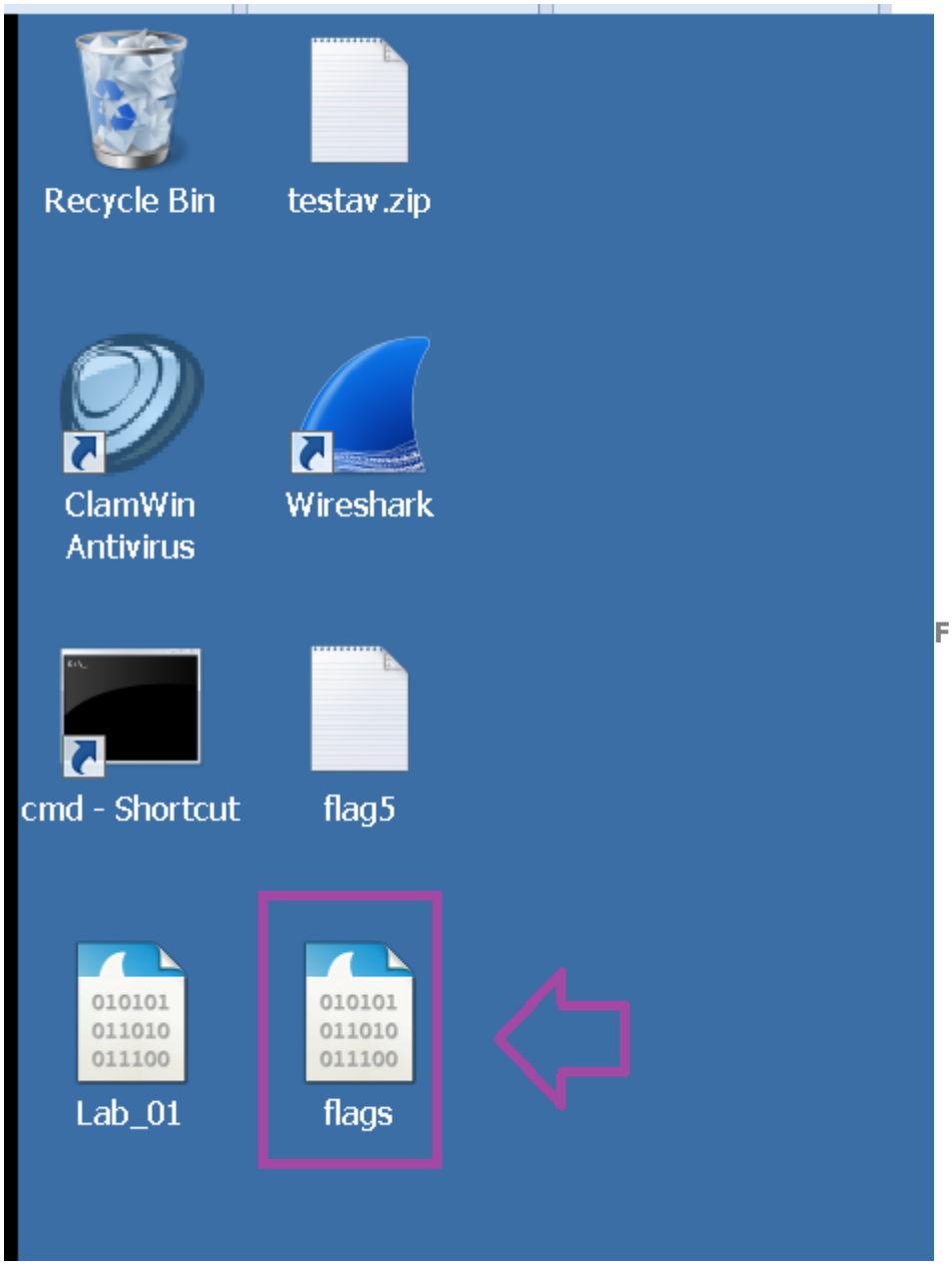

### OTHER INFORMATION – RAW FORMAT

7. **Close** the Lab_01 file by **clicking** the X in the top right hand pane of Wireshark.



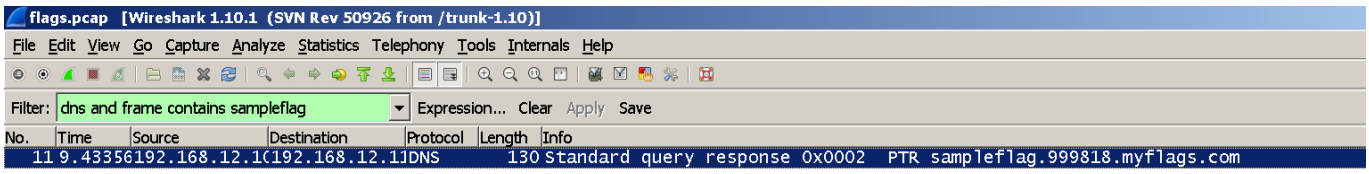

### CLOSE FILE

7. **Double-click** on the flags file on the desktop to open the Wireshark.




FLAGS FILE

7. In the filter pane, **type** `dns and frame contains sampleflag`

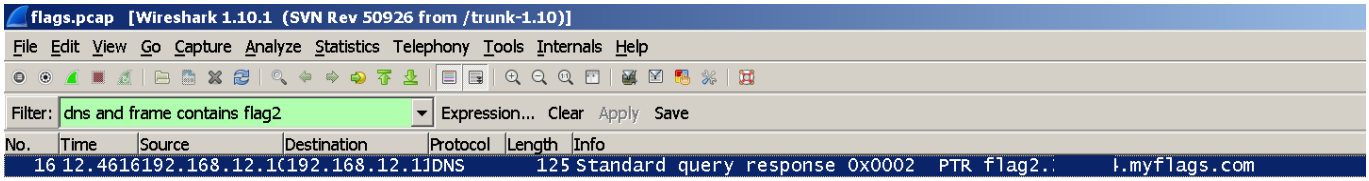



FILTER

**Notice** the flag of 999818. **Click** on the Challenge icon and **type** the flag number into the answer box. This is just to show you how to capture Challenge Flags you will see throughout this lab.

Challenge Sample #

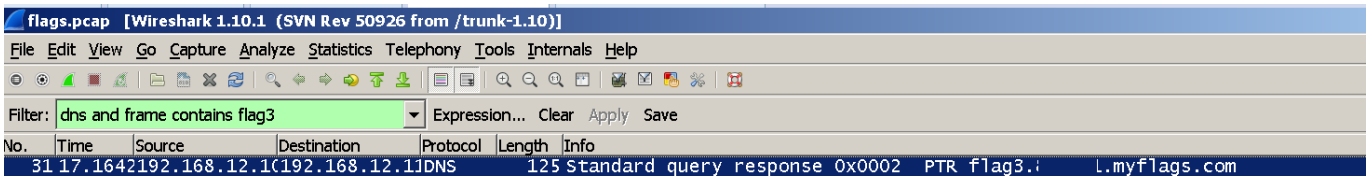
7. In the filter pane, **type** `dns and frame contains flag2`




**FILTER**

Challenge #

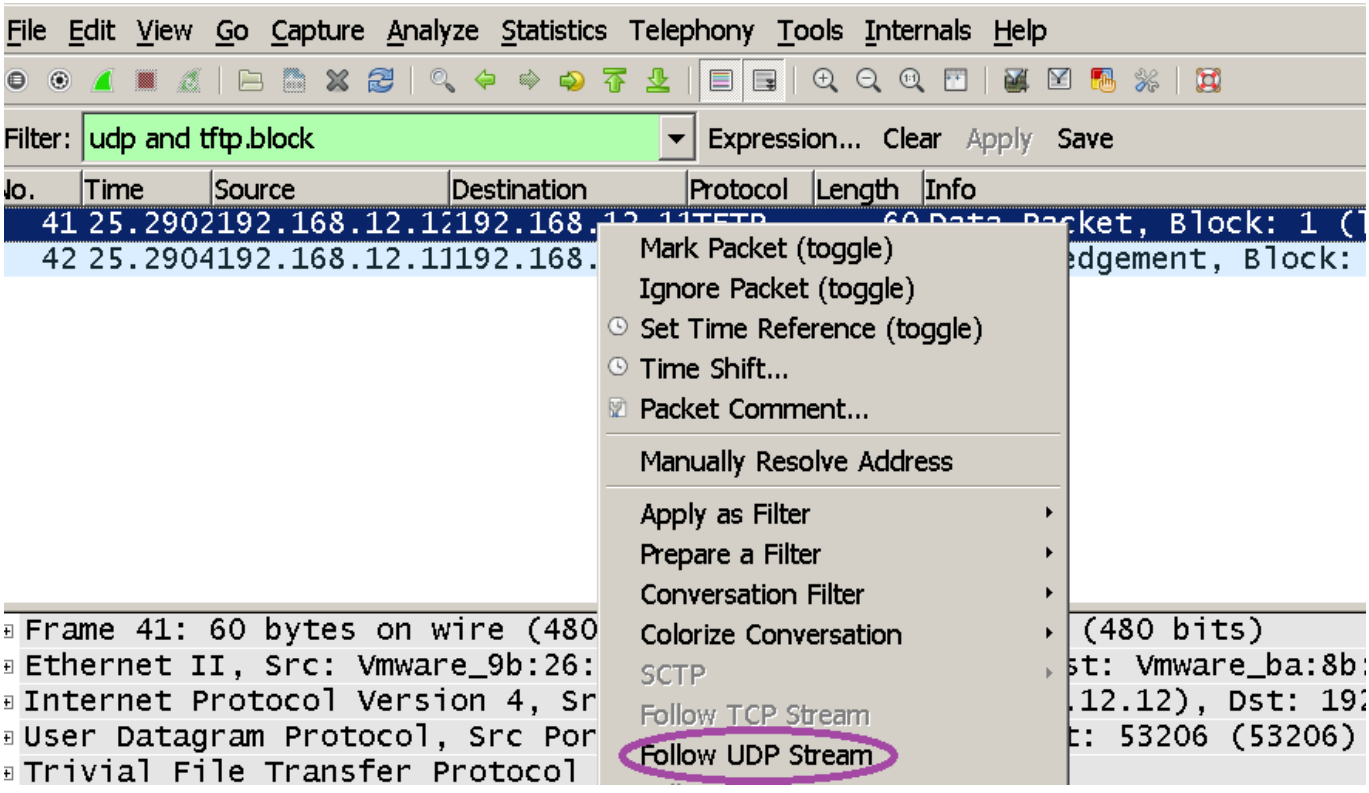
7. In the filter pane, **type** `dns and frame contains flag3`




**FILTER**

Challenge #

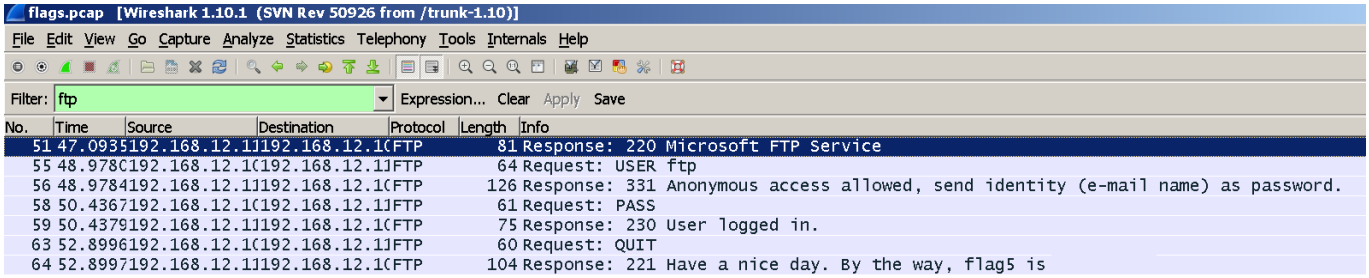
7. In the filter pane, **type** `udp and tftp.block`. **Right click** on the first frame and **select** follow UDP Stream




**FILTER**

Challenge #

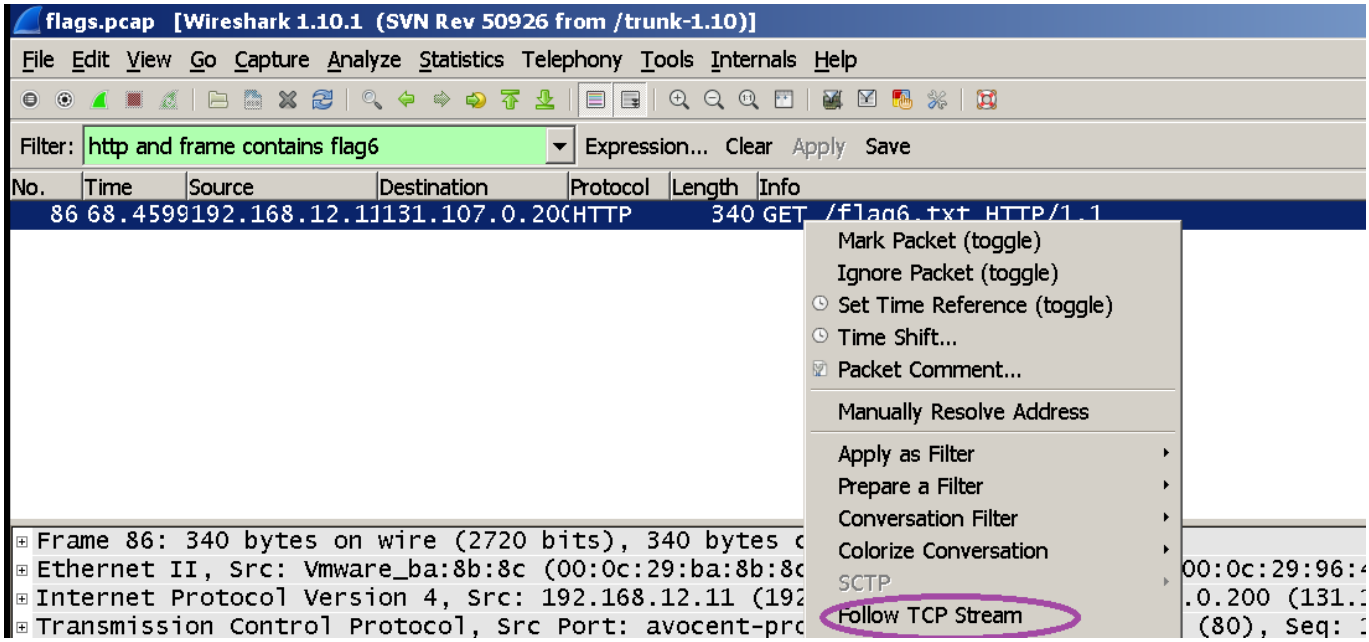
7. In the filter pane, **type** `ftp`.




**FILTER**

Challenge #

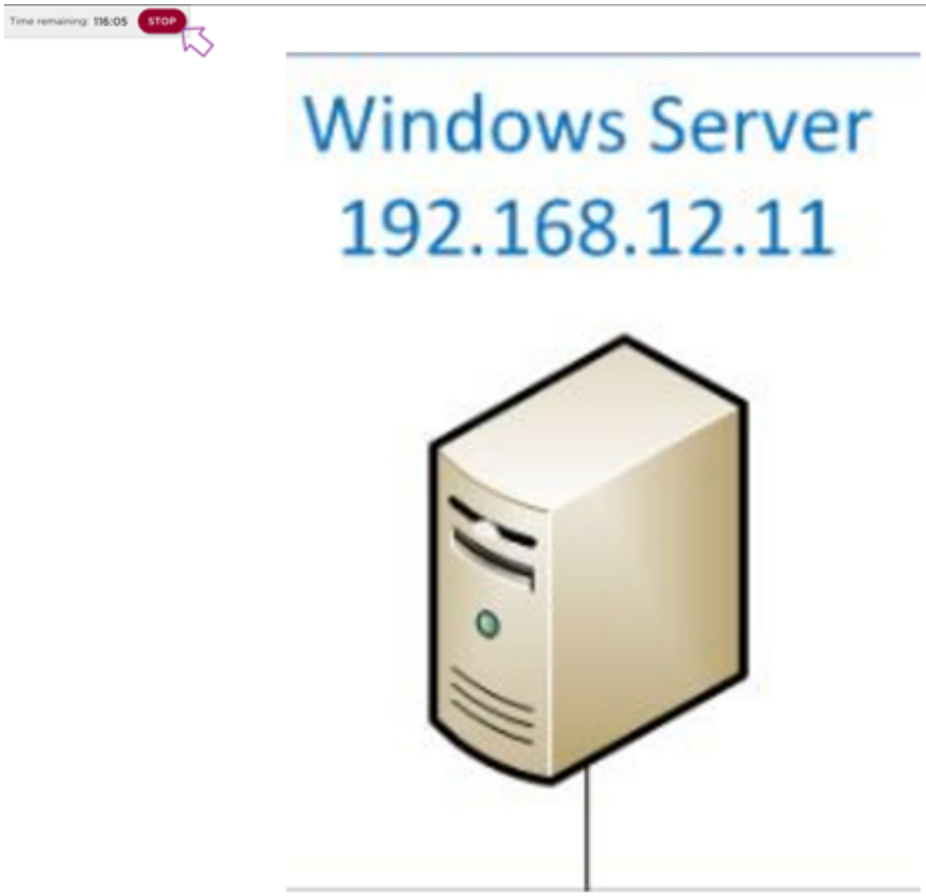
7. In the filter pane, **type** `http and frame contains flag6`. **Right click** on the first frame and **select** follow TCP Stream. **Scroll down** until you see the flag.




**FILTER**

Challenge #

Note: **Press** the STOP button to complete the lab.




**LAB COMPLETE**

## CONCLUSION:

The physical layer of the OSI model incorporates all of the tangible network components. This includes the electronic signals on the media that are used to represent the 1's and 0's that, in turn, make up the data you requested.

## DISCUSSION QUESTIONS:

1. What is the PDU associated with the physical layer of the OSI model?
2. True or false—a bad cable is considered a physical layer issue.
3. True or false—a misconfigured IP address is considered a physical layer issue.

# References:

**Organization:** Stanly Community College

Author: Trent Helms, Program Head, Networking Technology



The development of this document is funded by the Department of Labor (DOL) Trade Adjustment

Assistance Community College and Career Training (TAACCCT) Grant No. TC-22525-11-60-A-48. The National Information Security, Geospatial Technologies Consortium (NISGTC) is an entity of Collin College of Texas, Bellevue College of Washington, Bunker Hill Community College of Massachusetts, Del Mar College of Texas, Moraine Valley Community College of Illinois, Rio Salado College of Arizona, and Salt Lake Community College of Utah. 

---