WIRESHARKUNIVERSITY

# Wireshark® 101
# Essential Skills for Network Analysis
# Student Manual

2nd Edition

***Always ensure you have proper authorization before you listen to and capture network traffic.***

Protocol Analysis Institute, Inc
59 Damonte Ranch Parkway, #B340
Reno, Nevada 89521 USA
*www.packet-level.com*

Wireshark University
*info@wiresharktraining.com*
*www.wiresharktraining.com*

To arrange bulk purchase discounts for sales promotions, events, training courses, or other purposes, please contact Chappell University via email (*info@chappellU.com*), phone (1-408-378-7841), or mail (59 Damonte Ranch Parkway, #B340, Reno, NV 89521).

This Student Manual and course are based on the book, "*Wireshark 101: Essential Skills for Network Analysis*," available worldwide in print and electronic version. For more information, visit *www.wiresharkbook.com*.

Distributed worldwide for Chappell University through Protocol Analysis Institute, Inc. Protocol Analysis Institute, Inc. is the exclusive educational materials developer for Chappell University.

For general information on Chappell University or Protocol Analysis Institute, Inc., including information on corporate licenses, updates, future titles, or courses, contact the Protocol Analysis Institute, Inc., at 408/3787841 or send email to *info@chappellU.com*.

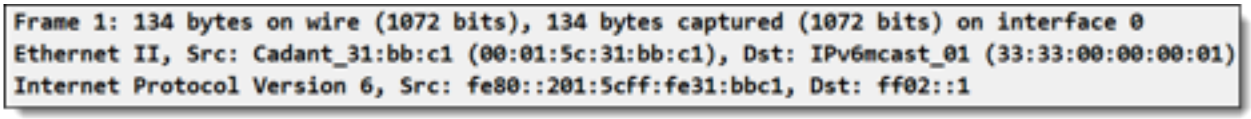
**Limit of Liability/Disclaimer of Warranty**. The author and publisher have used their best efforts in preparing this Student Manual and the related materials used in this Student Manual. Protocol Analysis Institute, Inc., Chappell University, and the author(s) make no representations or warranties of merchantability of fitness for a particular purpose. Protocol Analysis Institute, Inc., and Chappell University assume no liability for any damages caused by following the instructions or using the techniques or tools listed in this Student Manual or related materials used in this course. Protocol Analysis Institute, Inc., Chappell University, and the author(s) make no representations or warranties that extend beyond the descriptions contained in this paragraph. No warranty may be created or extended by sales representatives or written sales materials. The accuracy or completeness of the information provided herein and the opinions stated herein are not guaranteed or warranted to produce any particular result and the advice and strategies contained herein may not be suitable for every individual. Protocol Analysis Institute, Inc., Chappell University, and author(s) shall not be liable for any loss of profit or any other damages, including without limitation, special, incidental, consequential, or other damages.

# Lab 1: Use Packets to Build a Picture of a Network

When you are analyzing traffic, try to get a feel for the network layout from what you can learn in the packets. Who is sending the packets? Who are the targets? What are their MAC and IP addresses? If multiple hosts talk through a device, it is likely a router. Switches are transparent, but you must assume that clients go through switches to reach a router.

In this lab you will examine the MAC and IP addresses to build a picture of a portion of a network. In addition, you will look at the **Protocol** column to determine what applications are running on various hosts. Red text indicates that we just learned this information from the current frame.

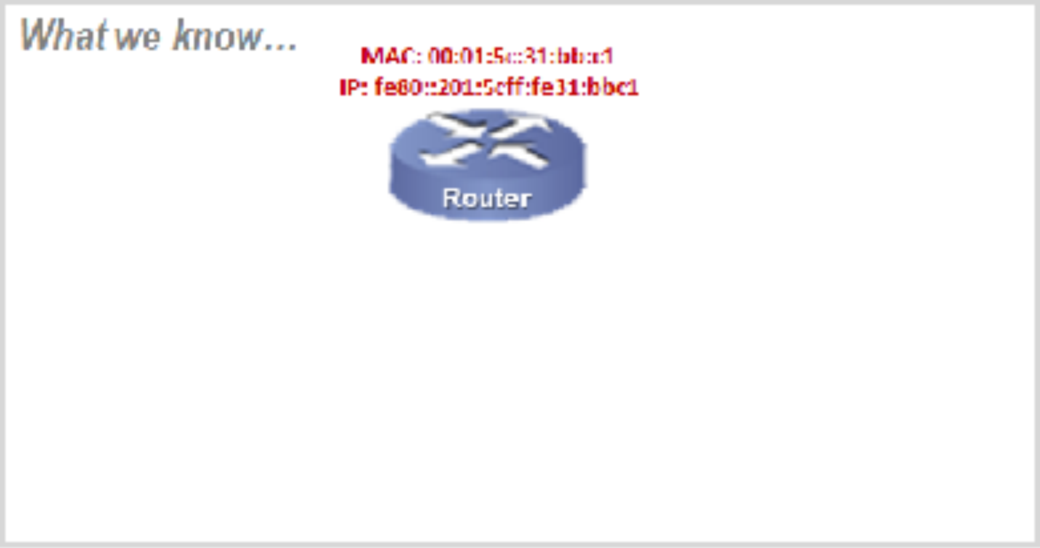
***Frame 1***



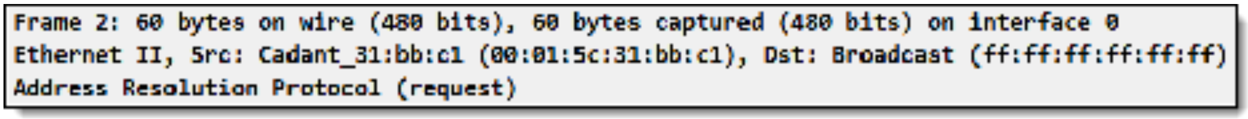
```
Frame 1: 134 bytes on wire (1072 bits), 134 bytes captured (1072 bits) on interface 0
Ethernet II, Src: Cadant_31:bb:c1 (00:01:5c:31:bb:c1), Dst: IPv6mcast_01 (33:33:00:00:00:01)
Internet Protocol Version 6, Src: fe80::201:5cff:fe31:bbc1, Dst: ff02::1
```

Launch **Wireshark**, click the **File Open** button on the main tool bar and double-click on ***general101.pcapng*** to open this file.

Examine the Packet List pane. Frame 1 uses IPv6. Look in the Ethernet and IP headers for this frame in the Packet Details pane (shown below). This appears to be an IPv6 multicast (note the *IPv6mcast* designation in the destination Ethernet address field).

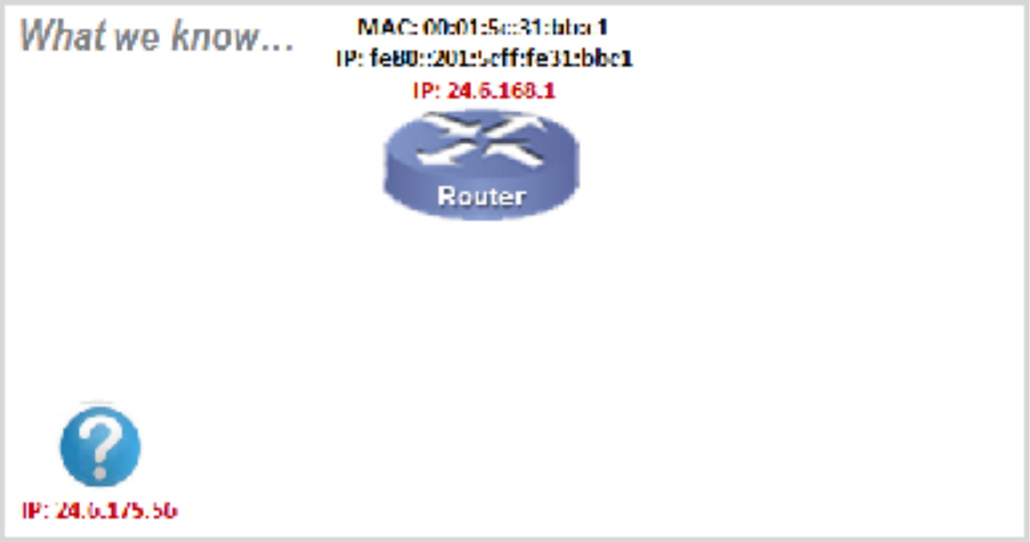


***Frame 2***

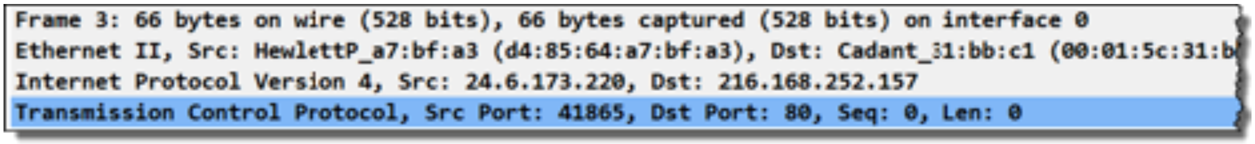


```
Frame 2: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
Ethernet II, Src: Cadant_31:bb:c1 (00:01:5c:31:bb:c1), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (request)
```

Frame 2 is an ARP packet. Look inside the Ethernet header then inside the ARP portion of the packet. This ARP request is sent to locate the MAC address of the Target IP Address.

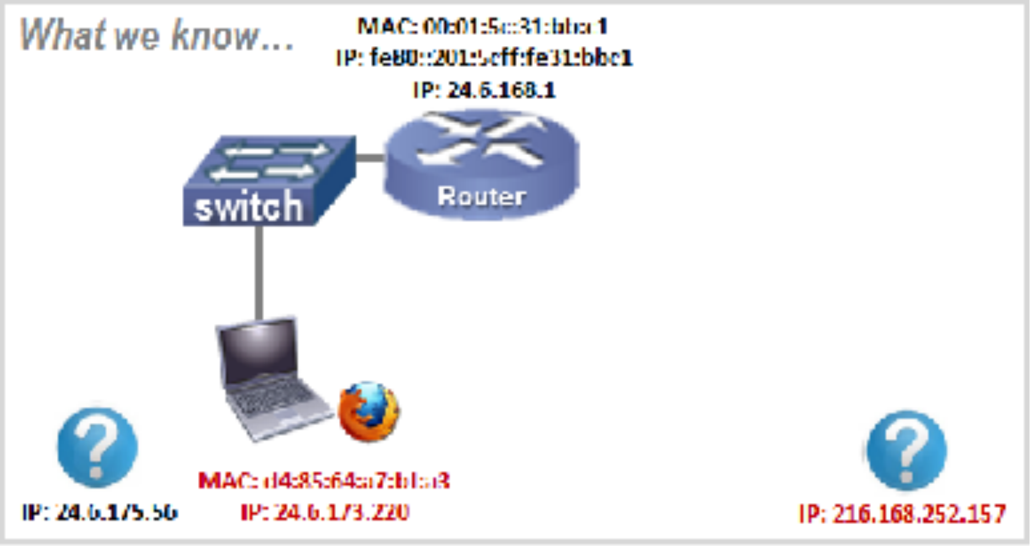


***Frame 3***

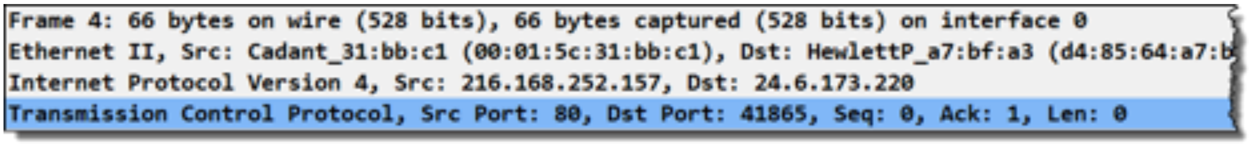


```
Frame 3: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: HewlettP_a7:bf:a3 (d4:85:64:a7:bf:a3), Dst: Cadant_31:bb:c1 (00:01:5c:31:b
Internet Protocol Version 4, Src: 24.6.173.220, Dst: 216.168.252.157
Transmission Control Protocol, Src Port: 41865, Dst Port: 80, Seq: 0, Len: 0
```

Frame 3 is a TCP handshake packet to the HTTP port. Again, look in the Ethernet header and IP header to build your picture of the network. Since the target has not responded, we really can't say the target is there. We will mark it with a question mark until we see it talk on the network.



***Frame 4***

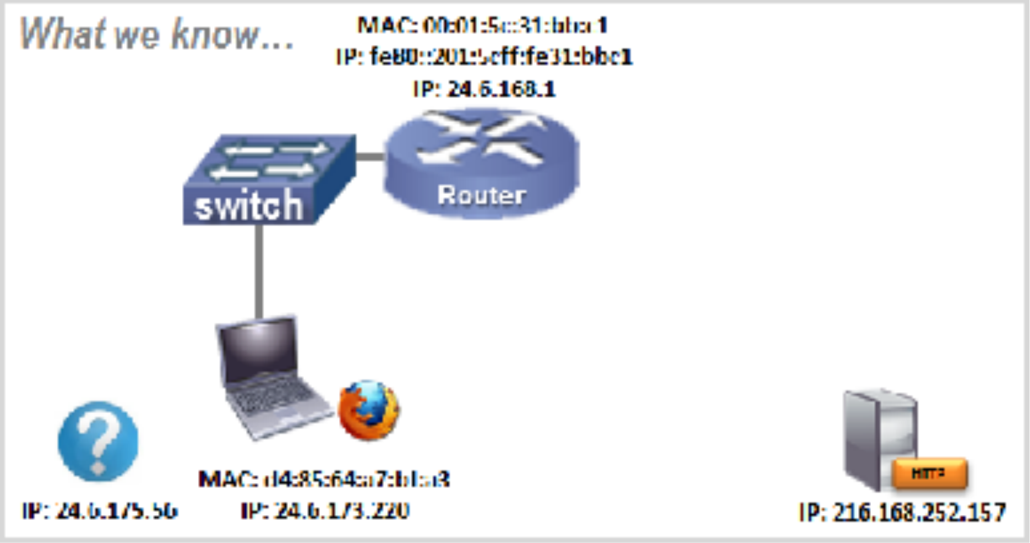


```
Frame 4: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: Cadant_31:bb:c1 (00:01:5c:31:bb:c1), Dst: HewlettP_a7:bf:a3 (d4:85:64:a7:b
Internet Protocol Version 4, Src: 216.168.252.157, Dst: 24.6.173.220
Transmission Control Protocol, Src Port: 80, Dst Port: 41865, Seq: 0, Ack: 1, Len: 0
```

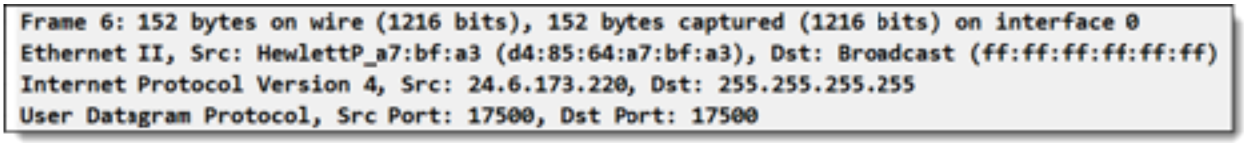
Frame 4 is the reply to frame 3. We can now draw in the new HTTP server in our diagram. Look at the source MAC address in frame 4. It comes from the router, not the source server.

Remember that routers strip off the received MAC header and apply a new MAC header. The new MAC header contains the address of the router's interface on this network as the new source MAC address and the address of the destination device as the new destination MAC address. This is how a router forwards a packet. On your local network, you may see traffic from many different IP addresses come from the MAC address of the local router.

Frame 5 finishes the TCP 3-way handshake.

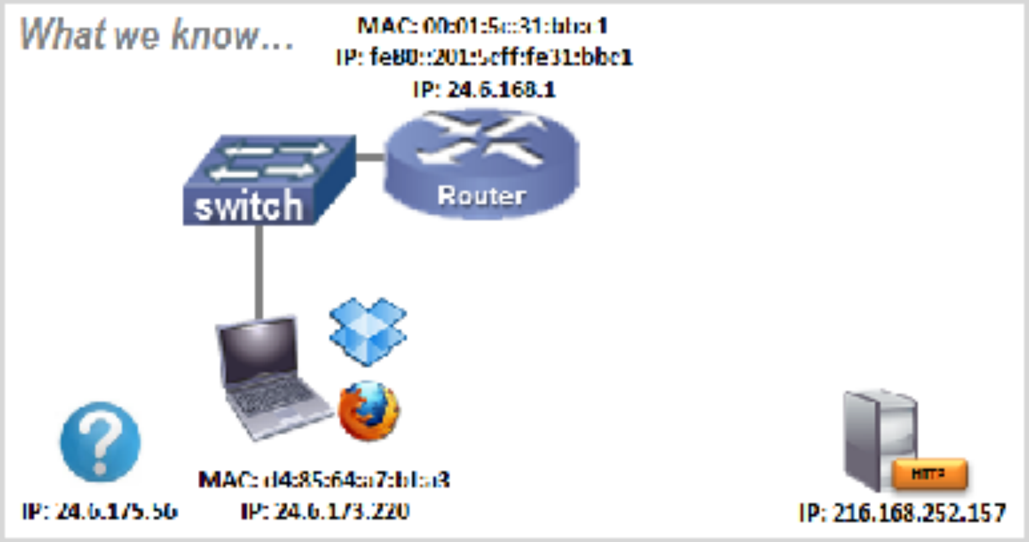


***Frame 6***

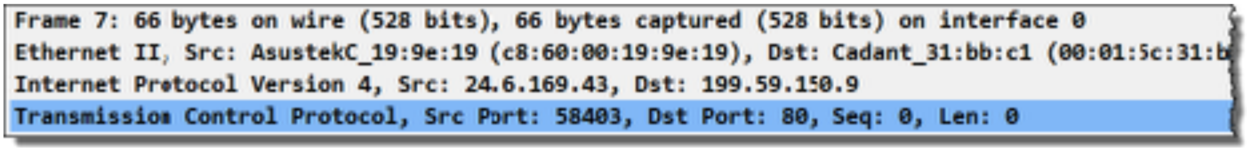


```
Frame 6: 152 bytes on wire (1216 bits), 152 bytes captured (1216 bits) on interface 0
Ethernet II, Src: HewlettP_a7:bf:a3 (d4:85:64:a7:bf:a3), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 24.6.173.220, Dst: 255.255.255.255
User Datagram Protocol, Src Port: 17500, Dst Port: 17500
```

Frame 6 is a Dropbox LAN Sync Discovery Protocol (DB-LSB-DISC) packet from our client. This packet is sent to the broadcast address.

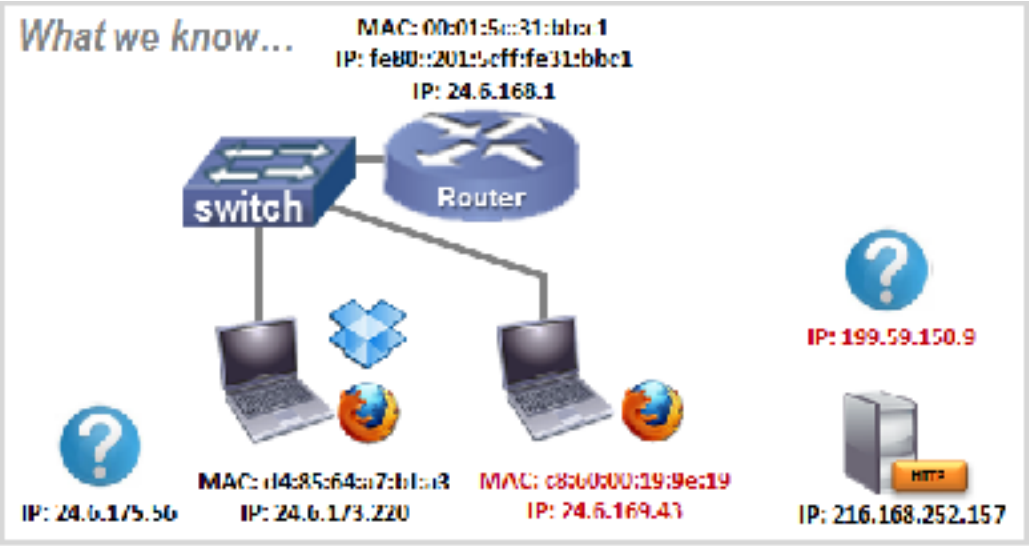


***Frame 7***

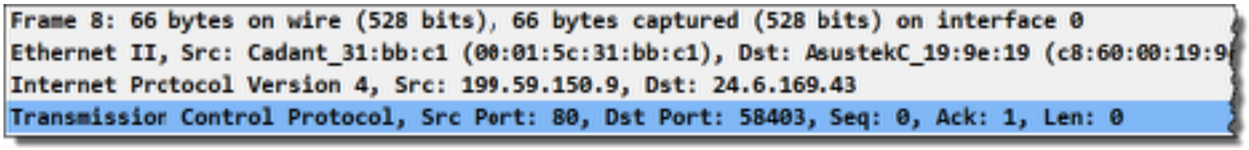


```
Frame 7: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: AsustekC_19:9e:19 (c8:60:00:19:9e:19), Dst: Cadant_31:bb:c1 (00:01:5c:31:b
Internet Protocol Version 4, Src: 24.6.169.43, Dst: 199.59.150.9
Transmission Control Protocol, Src Port: 58403, Dst Port: 80, Seq: 0, Len: 0
```

Frame 7 is another TCP handshake packet, but we have a new source and destination. We can now draw in a new source MAC and IP address and a new destination IP address. We must wait for the target to send a packet before we say it is definitely there.

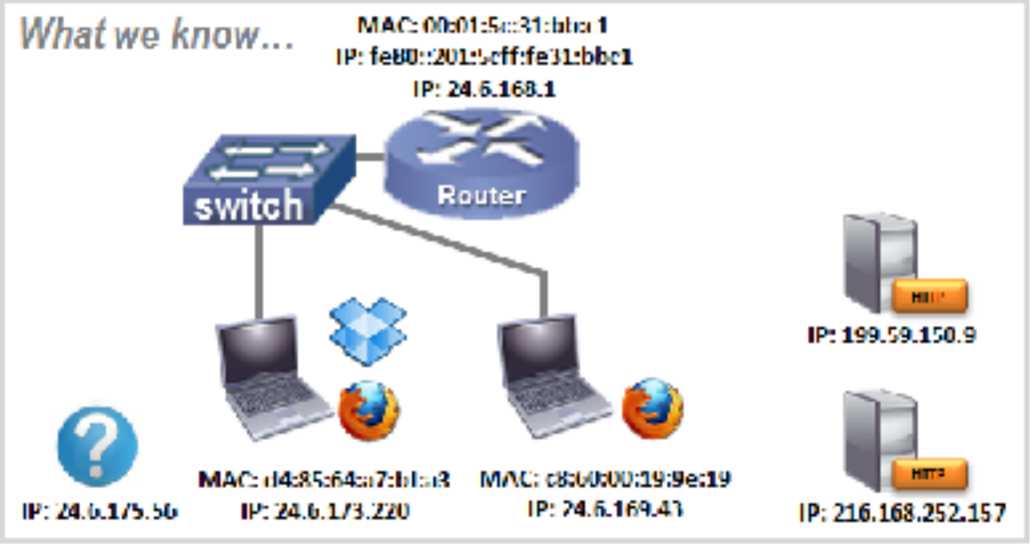


***Frame 8***

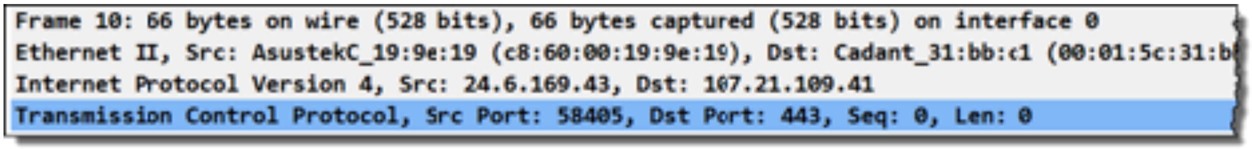


```
Frame 8: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: Cadant_31:bb:c1 (00:01:5c:31:bb:c1), Dst: AsustekC_19:9e:19 (c8:60:00:19:9
Internet Prctocol Version 4, Src: 199.59.150.9, Dst: 24.6.169.43
Transmission Control Protocol, Src Port: 80, Dst Port: 58403, Seq: 0, Ack: 1, Len: 0
```

Frame 8 is the answer from the HTTP server (199.59.150.9). We now know that this server is talking on the wire. Frame 9 is the final piece of the TCP handshake.

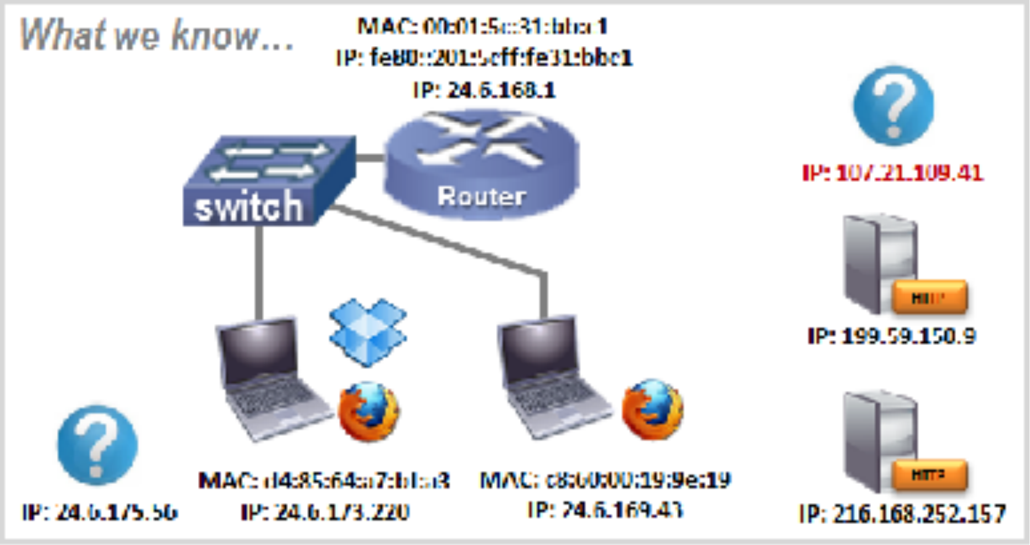


***Frame 10***

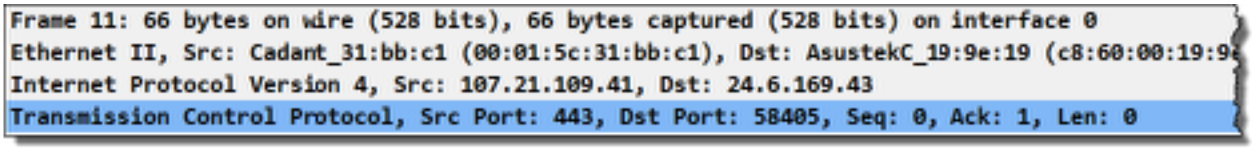


```
Frame 10: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: AsustekC_19:9e:19 (c8:60:00:19:9e:19), Dst: Cadant_31:bb:c1 (00:01:5c:31:b
Internet Protocol Version 4, Src: 24.6.169.43, Dst: 107.21.109.41
Transmission Control Protocol, Src Port: 58405, Dst Port: 443, Seq: 0, Len: 0
```

Frame 10 indicates that the other local host is trying to connect to another server. This time the target is port 443, the HTTPS port.

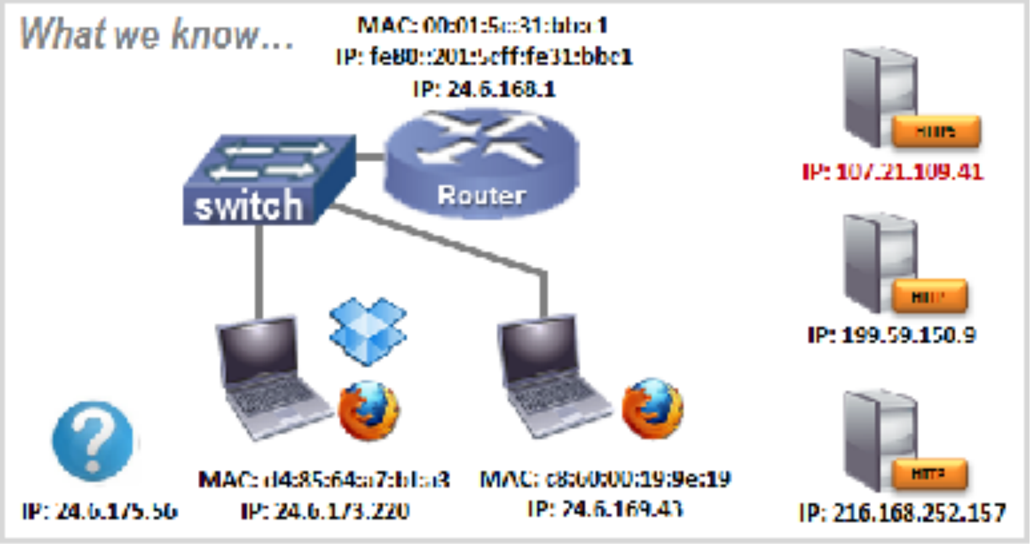


***Frame 11***



```
Frame 11: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: Cadant_31:bb:c1 (00:01:5c:31:bb:c1), Dst: AsustekC_19:9e:19 (c8:60:00:19:9
Internet Protocol Version 4, Src: 107.21.109.41, Dst: 24.6.169.43
Transmission Control Protocol, Src Port: 443, Dst Port: 58405, Seq: 0, Ack: 1, Len: 0
```

Frame 11 is a response from the target. We can now assume the target is running. Frame 12 finishes the TCP handshake and our drawing of the network we discovered just by looking at these first few packets in the trace file.



As you can see, lots of different conversations are occurring simultaneously. We can build a picture of the network based on the packets we see. Building an image of a network based on traffic is a common task used in analysis.

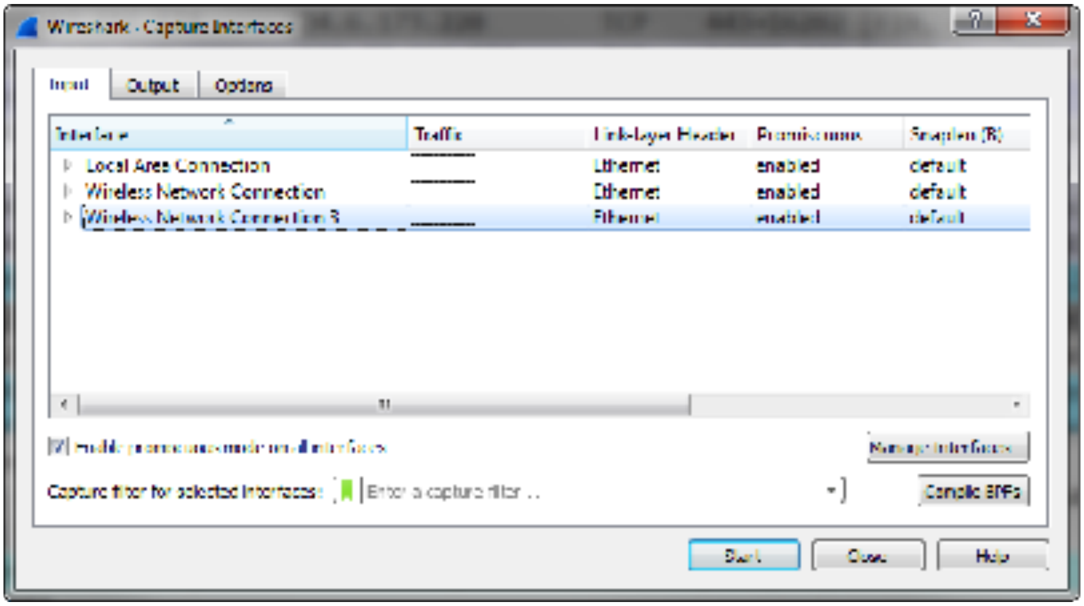
## Lab 2: Capture and Classify Your Own Background Traffic (Optional) – NOTE Sparklines not Working

Take a moment and capture your own background as we did in this section. When you complete your capture, perform some research on the resulting trace file to see if you can characterize all the traffic to/from your machine when you are not touching the keyboard.

**Step 1**: Close all applications except for Wireshark and any normal background applications that run on your machine.

**Step 2:** Click the **Capture Options** button on the main toolbar.

**Step 3**: Select the interface that sees active traffic.



Wireshark · Capture Interfaces

| Interface | Traffic | Link-layer Header | Promiscuous | Snaplen (B) |
|---|---|---|---|---|
| Local Area Connection | | Ethernet | enabled | default |
| Wireless Network Connection | | Ethernet | enabled | default |
| Wireless Network Connection 3 | | Ethernet | enabled | default |

☑ Enable promiscuous mode on all interfaces

Capture filter for selected Interfaces: Enter a capture filter …

**Step 4**: Click **Start**. Let the capture run for at least five minutes (longer if you can wait).

**Step 5**: Click the **Stop Capture** button 🟥 on the main toolbar.

Spend some time going through the trace file to identify the applications that run in the background on your machine. Focus on the **Protocol** column and the **Info** column.

If you don't recognize the application, perform some research on the IP addresses that your system communicates with. Most likely you will also see broadcast or multicast traffic from other hosts on your network.

**Step 6**: To save this file, click the **Save** button on the main toolbar, navigate to the target directory, and name your file ***background1.pcapng***.

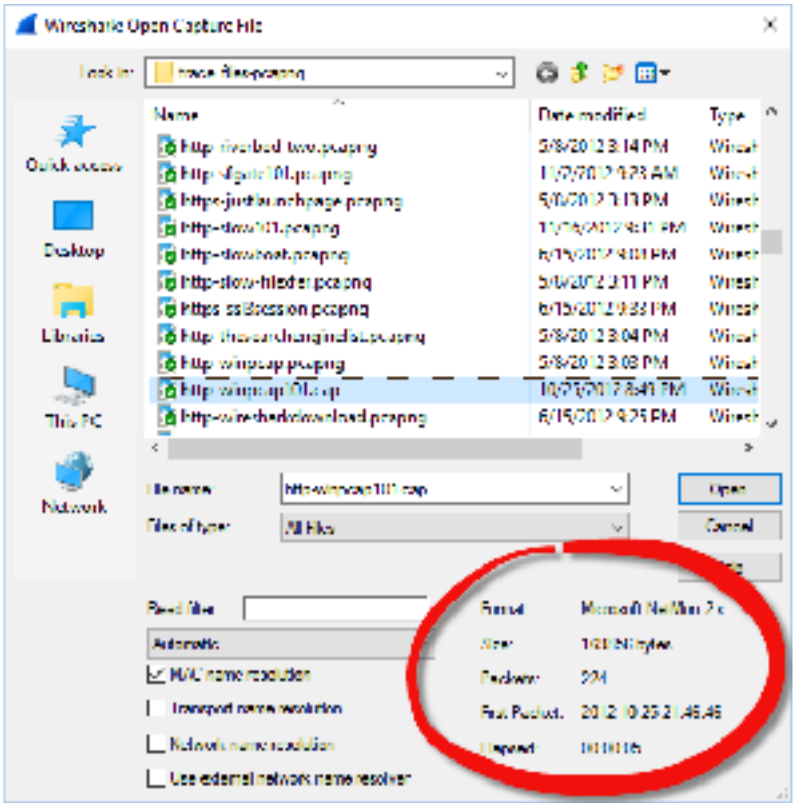
Recognizing your own background traffic will help you remove this from consideration when looking for unusual communications. Consider saving trace files of your "normal" traffic to refer to when troubleshooting.

## Lab 3: Open a Network Monitor .cap File

In this lab you will use Wireshark's Wiretap Library to open a file captured with Microsoft's Network Monitor[1].

**Step 1**: Click the **File Open** button on the main toolbar.

**Step 2**: Navigate to your trace file directory and click on ***http-winpcap101.cap***. Wireshark looks inside the trace file to identify what tool was used to capture the traffic, as shown below. Although this file was captured with Microsoft's Network Monitor (NetMon) v3.4, Wireshark marks it as NetMon v2 because that is the format v3.4 saves in.



Wireshark Open Capture File

Look in: trace files-pcapng

| Name | Date modified | Type |
|---|---|---|
| http-riverbed-two.pcapng | 5/8/2012 3:14 PM | Wiresh |
| http-sfgate101.pcapng | 11/7/2012 9:23 AM | Wiresh |
| https-justlaunchpage.pcapng | 5/8/2012 3:13 PM | Wiresh |
| http-slow101.pcapng | 11/16/2012 [illegible] PM | Wiresh |
| http-slowboot.pcapng | 6/15/2012 [illegible] PM | Wiresh |
| http-slow-filexfer.pcapng | 5/8/2012 3:11 PM | Wiresh |
| https-ssl3session.pcapng | 6/15/2012 9:33 PM | Wiresh |
| http-[illegible].pcapng | 5/8/2012 3:04 PM | Wiresh |
| http-winpcap.pcapng | 5/8/2012 3:08 PM | Wiresh |
| http-winpcap101.cap | 10/25/2012 8:40 PM | Wiresh |
| http-wiresharkdownload.pcapng | 6/15/2012 9:25 PM | Wiresh |

File name: http-winpcap101.cap

Files of type: All Files

Read filter:

Automatic

☑ MAC name resolution

□ Transport name resolution

□ Network name resolution

□ Use external network name resolver

Format: Microsoft NetMon 2.x

Size: 161 KiBytes

Packets: 224

First Packet: 2012-10-25 21:46:46

Elapsed: 00:00:16

**Step 3**: Click **Open**. Once the file is open, select **File | Save As** and click the drop down menu arrow next to Files of Type. Select **Wireshark – pcapng (*pcapng;*.pcapng.gz;*.ntar;*.ntar.gz)** and name the file ***httpwinpcap101.pcapng***.

Wireshark can recognize and open trace files created with most other industry tools. Once open, the fact that this trace file was captured with Network Monitor is transparent to you.

[1] Microsoft Network Monitor was replaced with Microsoft Message Analyzer, but Message Analyzer can still save trace files in the native Network Monitor .cap format.

## Lab 4: Add the HTTP Host Field as a Column

During a browsing session, an HTTP client sends requests for HTTP objects to one or more HTTP servers. In each of the requests, the client specifies the name or the IP address of the target HTTP server. This can be very revealing.

***Note**: All frames from 24.6.173.220 will appear with a black background and red foreground if Wireshark is set to validate IP header checksums. You will ensure this feature is disabled in Lab 5.*

**Step 1**: Click the **File Open** button on the main toolbar and open ***httpdisney101.pcapng***.

**Step 2**: First we will hide the Time to Live column (if you created one while following along with the previous section of this book). Right-click the **Time to Live** column heading and uncheck that column in the drop-down menu. If you want to see that column again later, simply right-click on any column heading and click it in the column list to enable it.

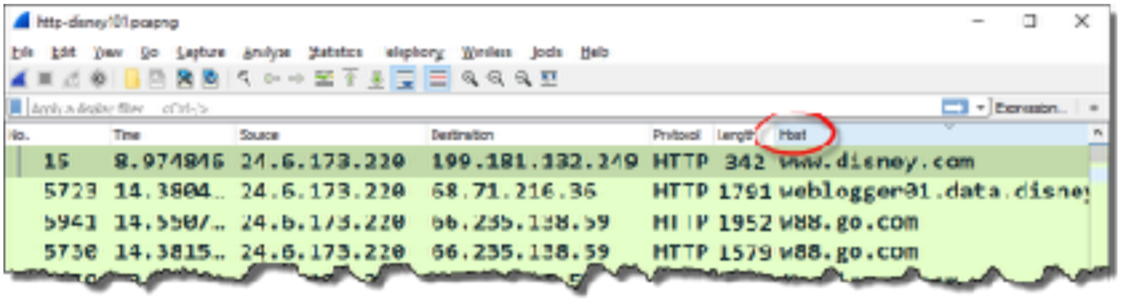
**Step 3**: Scroll down in the Packet List pane and select **frame 15**.

**Step 4**: The Packet Details pane shows the contents of frame 15. Click the › in front of Hypertext Transfer Protocol to expand this section of the frame.

**Step 4**: Right-click on the **Host** line (which contains `www.disney.com\r\n`) and select **Apply as Column**. Your new **Host** column appears to the left of the **Info** column. You can click and drag the right-hand edge of the column to widen or narrow the column.

**Step 5**: Click on the **Host** column heading twice to sort the column from high to low.

**Step 6**: Click the **Go to First** button to jump to the top of the sorted trace file. You can now easily see all the hosts to which the client sent requests, as shown below.


http-disney101.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter <Ctrl-/>
Expression...

| No. | Time | Source | Destination | Protocol | Length | Host |
|---|---|---|---|---|---|---|
| 15 | 8.974846 | 24.6.173.220 | 199.181.132.249 | HTTP | 342 | www.disney.com |
| 5723 | 14.3804.. | 24.6.173.220 | 68.71.216.36 | HTTP | 1791 | weblogger01.data.disney |
| 5941 | 14.5507.. | 24.6.173.220 | 66.235.138.59 | HTTP | 1952 | w88.go.com |
| 5730 | 14.3815.. | 24.6.173.220 | 66.235.138.59 | HTTP | 1579 | w88.go.com |

**Step 7**: Lab Clean-up Right-click on your new **Host** column heading and select **Hide Column**. If you want to view this column again, right-click any column heading and select **Displayed Columns | Host (*http.host*)**.

Adding and sorting columns are two key tasks that can shorten your analysis time significantly. Why go searching through thousands of packets when you can have Wireshark quickly gather and display the information you need?

## Lab 5: Set Key Wireshark Preferences (IMPORTANT LAB)[2]

Wireshark offers several key preference settings to enhance your analysis sessions. In this lab you will use **Edit | Preferences** on the main menu and the right-click method to view and change the preference settings.

These are the settings we will view and alter in this lab:

- Display filters that Wireshark will remember
- Recently opened files that Wireshark will remember
- Ethernet, IP, UDP, and TCP checksum validations
- TCP Calculate conversation timestamps setting
- TCP Track number of bytes in flight setting
- TCP Allow subdissector to reassemble TCP streams setting

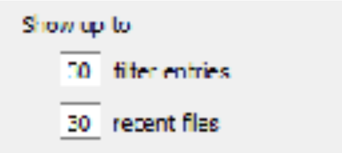
**Note**: Your Wireshark system should retain all of these settings through the rest of this course with the exception of the TCP *Allow subdissector to reassemble TCP streams* setting, which you will work with during various labs.

**Step 1**: Open ***http-pcaprnet101.pcapng***.

**Step 2:** Select **Edit | Preferences** on the main menu.

**Step 3:** Change both the **filter entries** and **recent files** settings to **30**.

These two settings allow you to quickly recall more of your recent filter settings and opened files.



Show up to

30 filter entries

30 recent files

**Step 4:** Click **OK**. This automatically applies and saves your settings in this *Default* profile and closes the Preferences window.

Next we will use the right-click method to check and change the Ethernet, IP, UDP, and TCP settings.

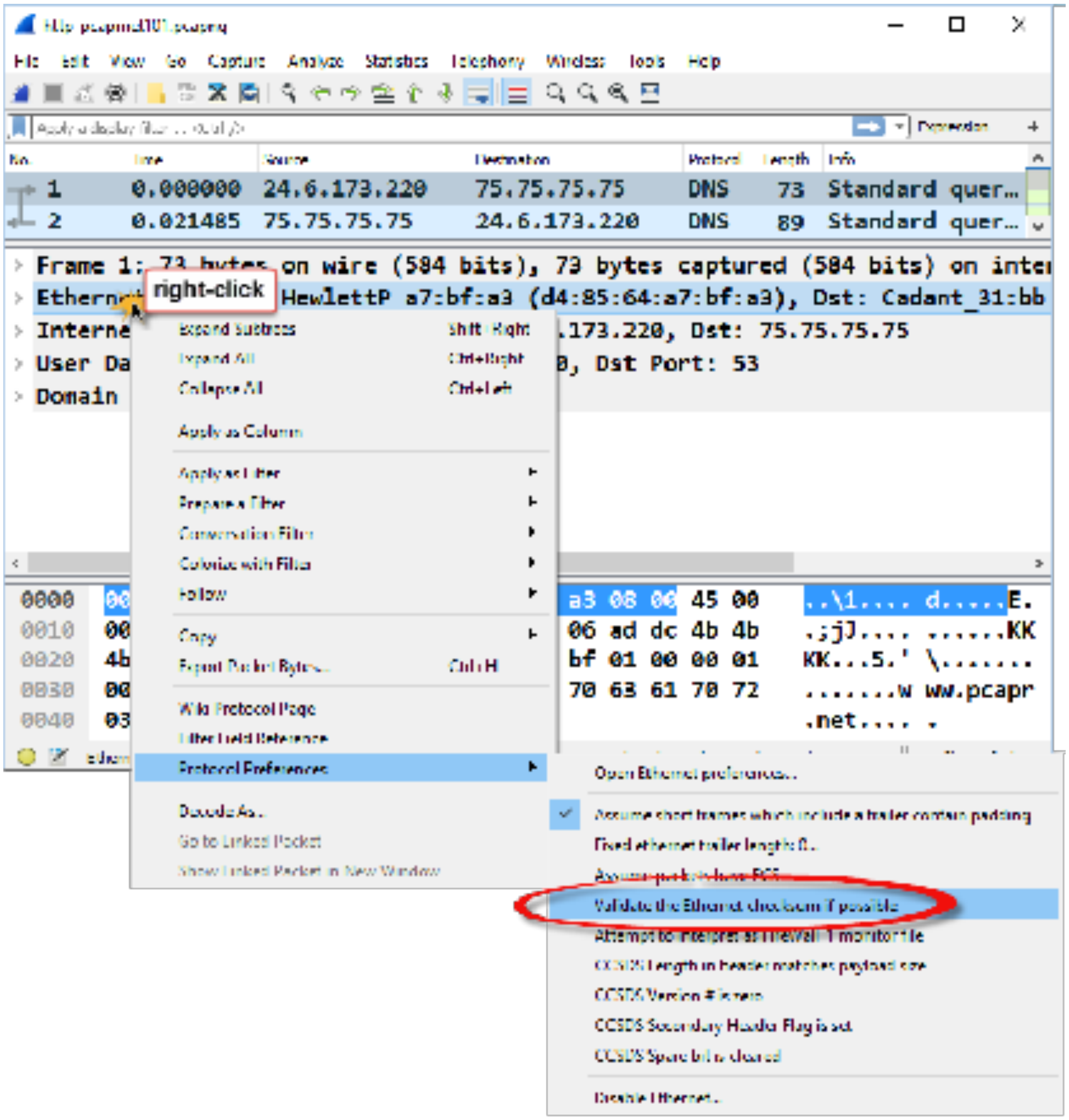
We will begin by disabling the Ethernet checksum validation (which is enabled by default).

Next, we will ensure IP, UDP, and TCP checksum validations are disabled[3]. These three checksum validations should already be disabled unless you updated Wireshark while retaining previous settings.

[2] The remaining labs in this course assume you have successfully completed this lab.

[3] Most systems support checksum offloading. If Wireshark obtains a copy of an outbound frame before the checksum values have been calculated, it will mark the checksums invalid. This is a false positive when capturing traffic directly on a host that supports checksum offloading.

**Step 5:** With frame 1 selected in the Packet List pane, right-click on the **Ethernet II** section of the Packet Details pane and hover over the **Protocol Preferences** option on the drop-down menu. If this setting is enabled (checked), click on the **Validate the Ethernet checksum if possible** setting to disable it.

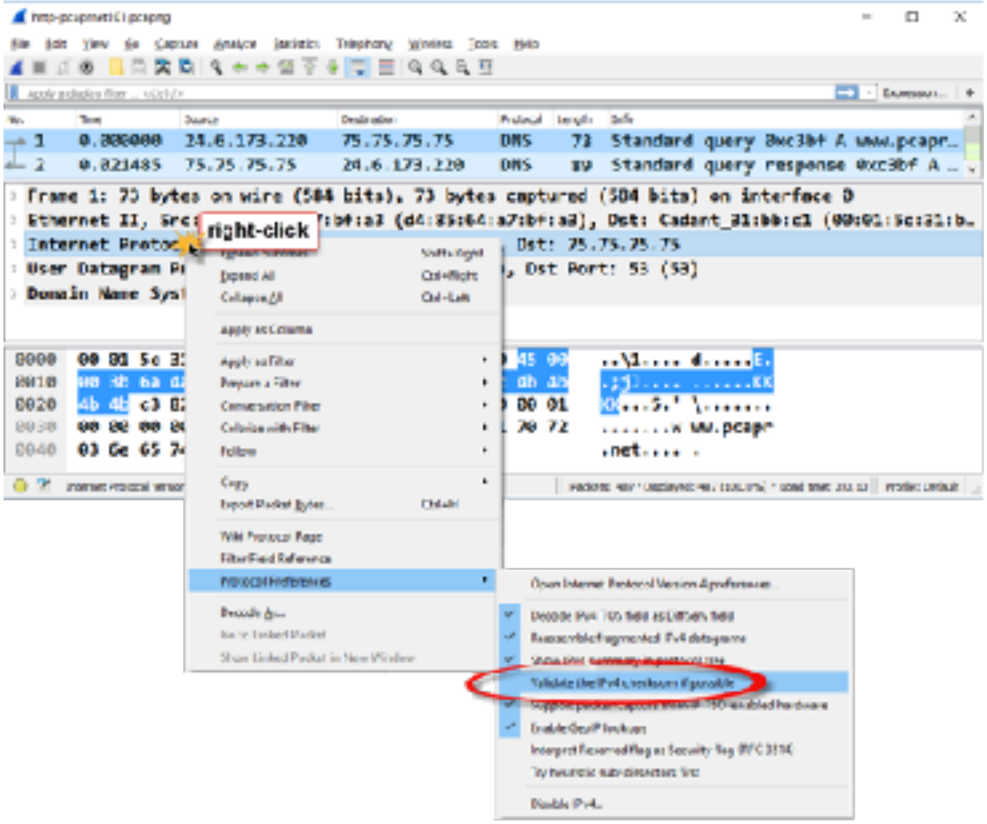

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
right-click
Expand Subtrees Shift+Right
Expand All Ctrl+Right
Collapse All Ctrl+Left
Apply as Column
Apply as Filter
Prepare a Filter
Conversation Filter
Colorize with Filter
Follow
Copy
Export Packet Bytes... Ctrl+H
Wiki Protocol Page
Filter Field Reference
Protocol Preferences
Decode As...
Go to Linked Packet
Show Linked Packet in New Window
Open Ethernet preferences...
Assume short frames which include a trailer contain padding
Fixed ethernet trailer length: 0...
Assume packets have FCS
Validate the Ethernet checksum if possible
Attempt to interpret as FireWall-1 monitor file
CCSDS Length in header matches payload size
CCSDS Version # is zero
CCSDS Secondary Header Flag is set
CCSDS Spare bit is cleared
Disable Ethernet...

```
No.   Time       Source         Destination    Protocol Length Info
1     0.000000   24.6.173.220   75.75.75.75    DNS      73     Standard quer…
2     0.021485   75.75.75.75    24.6.173.220   DNS      89     Standard quer…
```

```
> Frame 1: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on inte
> Etherne          HewlettP a7:bf:a3 (d4:85:64:a7:bf:a3), Dst: Cadant_31:bb
> Interne          .173.220, Dst: 75.75.75.75
> User Da          0, Dst Port: 53
> Domain
```

```
0000  00          a3 08 00 45 00    ..\1.... d.....E.
0010  00          06 ad dc 4b 4b    .;jJ.... ......KK
0020  4b          bf 01 00 00 01    KK...5.' \.......
0030  00          70 63 61 70 72    .......w ww.pcapr
0040  03                            .net.... .
```

**Step 6:** With frame 1 still selected in the Packet List pane, right-click on the **Internet Protocol** section of the Packet Details pane and hover over the **Protocol Preferences** option on the drop-down menu. If this setting is enabled (checked), click on the **Validate the IPv4 checksum if possible** setting to disable it.



```
1   0.000000   24.6.173.220   75.75.75.75    DNS   73   Standard query 0xc3bf A www.pcapr…
2   0.021485   75.75.75.75    24.6.173.220   DNS   89   Standard query response 0xc3bf A …
Frame 1: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface 0
Ethernet II, Src: …:bf:a3 (d4:85:64:a7:bf:a3), Dst: Cadant_31:bb:c1 (00:01:5c:31:b…
Internet Protoc… Dst: 75.75.75.75
User Datagram Pr… Dst Port: 53 (53)
Domain Name Syst…
```

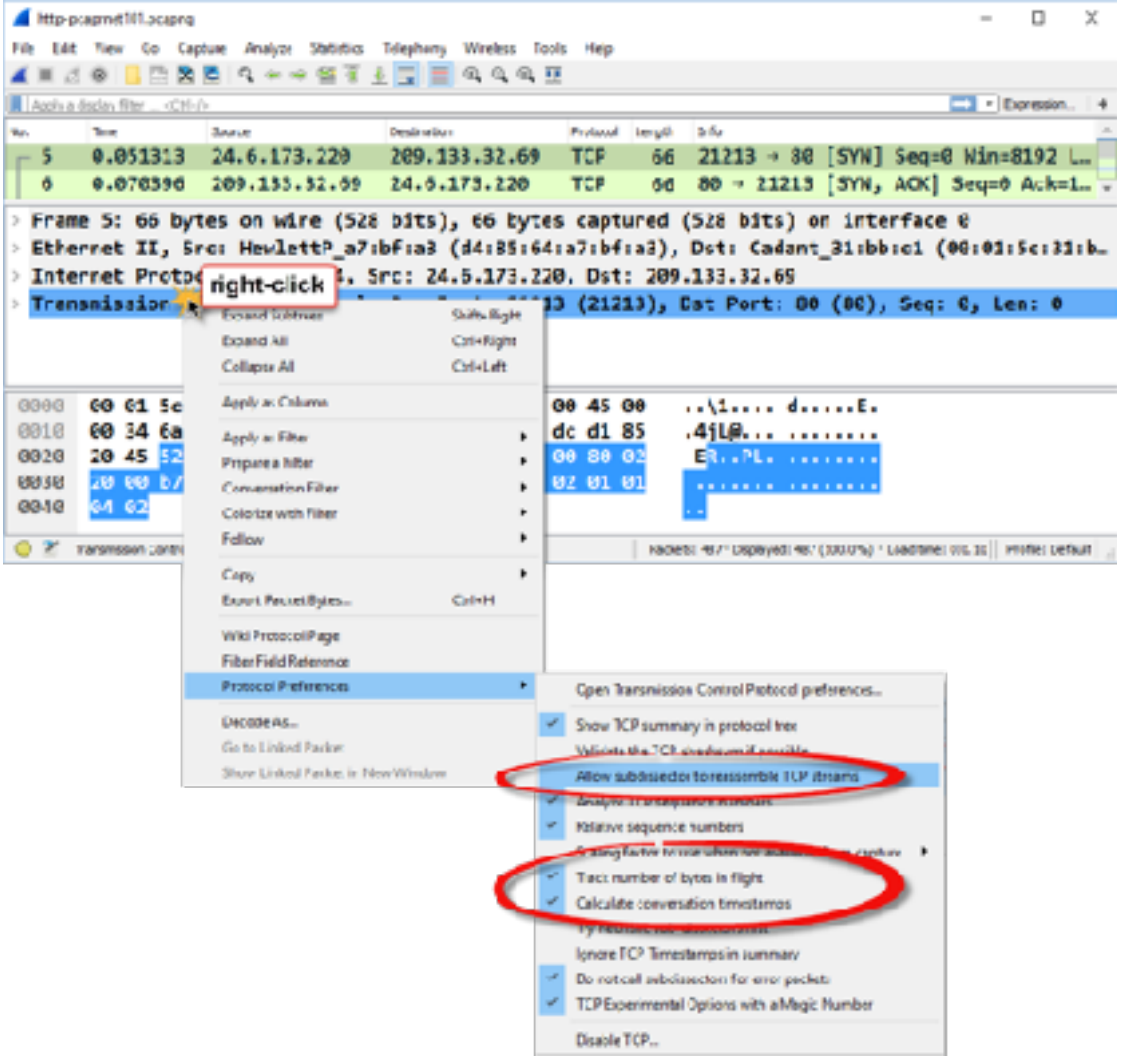
right-click
Expand All Ctrl+Right
Collapse All Ctrl+Left
Apply as Column
Apply as Filter
Prepare a Filter
Conversation Filter
Colorize with Filter
Follow
Copy
Export Packet Bytes… Ctrl+H
Wiki Protocol Page
Filter Field Reference
Protocol Preferences
Decode As…
Go to Linked Packet
Show Linked Packet in New Window
Open Internet Protocol Version 4 preferences…
Decode IPv4 TOS field as DiffServ field
Reassemble fragmented IPv4 datagrams
Show IPv4 summary in protocol tree
Validate the IPv4 checksum if possible
Support packet-capture from IP TSO-enabled hardware
Enable GeoIP lookups
Interpret Reserved flag as Security flag (RFC 3514)
Try heuristic sub-dissectors first
Disable IPv4…

**Step 7:** Again, in frame 1, right-click the **User Datagram Protocol** section of the Packet Details pane and hover over the **Protocol Preferences** option from the drop-down menu. Uncheck **Validate the UDP checksum if possible** setting if it is currently enabled.

**Step 8:** Select **frame 5** in the Packet List pane. Right-click the **Transmission Control Protocol** section of the Packet Details pane and, under **Protocol Preferences**, disable the **Validate the TCP checksum if possible** setting if it is currently enabled.

**Step 9:** Since Wireshark closes the TCP protocol settings menu after you select an option, you must right-click again on the **Transmission Control Protocol** section of the Packet Details pane to review or change the following additional settings.

- Disabled: ***Allow subdissector to reassemble TCP streams***
- Enabled: ***Track number of bytes in flight***
- Enabled: ***Calculate conversation timestamps***

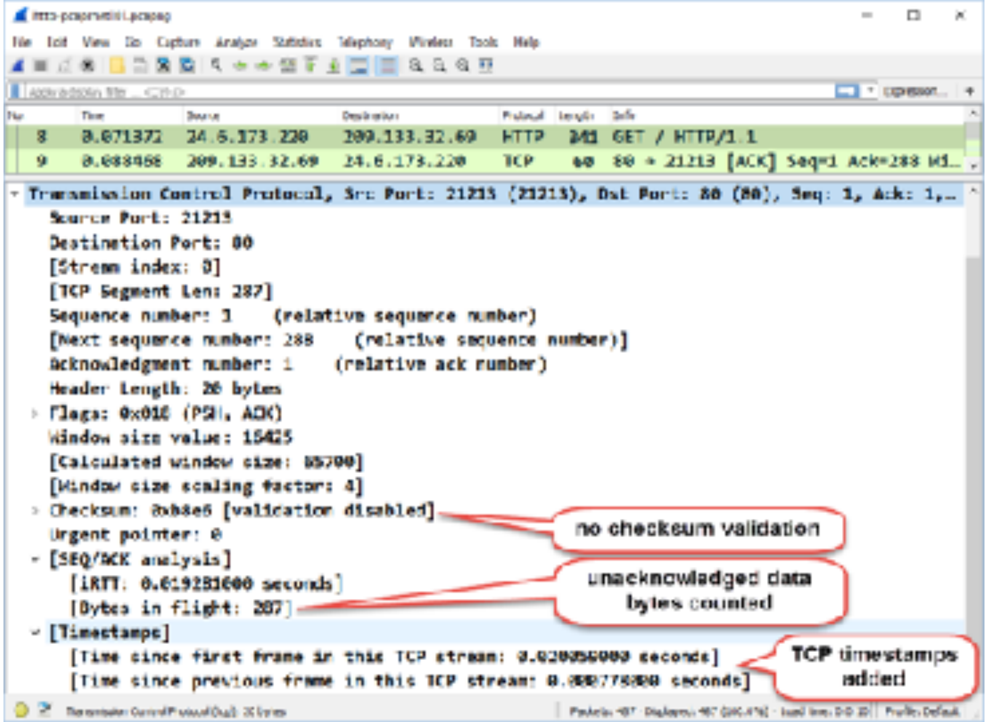


```
5  0.051313  24.6.173.220   209.133.32.69  TCP  66  21213 → 80 [SYN] Seq=0 Win=8192 L…
6  0.070396  209.133.32.69  24.6.173.220   TCP  66  80 → 21213 [SYN, ACK] Seq=0 Ack=1…

> Frame 5: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
> Ethernet II, Src: HewlettP_a7:bf:a3 (d4:85:64:a7:bf:a3), Dst: Cadant_31:bb:c1 (00:01:5c:31:b…
> Internet Proto… Src: 24.6.173.220, Dst: 209.133.32.69
> Transmission … (21213), Dst Port: 80 (80), Seq: 0, Len: 0
```

http-pcapnet101.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Expression…
right-click
Expand Subtrees Shift+Right
Expand All Ctrl+Right
Collapse All Ctrl+Left
Apply as Column
Apply as Filter
Prepare a Filter
Conversation Filter
Colorize with Filter
Follow
Copy
Export Packet Bytes… Ctrl+H
Wiki Protocol Page
Filter Field Reference
Protocol Preferences
Decode As…
Go to Linked Packet
Show Linked Packet in New Window
Open Transmission Control Protocol preferences…
✓ Show TCP summary in protocol tree
Validate the TCP checksum if possible
Allow subdissector to reassemble TCP streams
✓ Analyze TCP sequence numbers
✓ Relative sequence numbers
Scaling factor to use when not available from capture
✓ Track number of bytes in flight
✓ Calculate conversation timestamps
Try heuristic sub-dissectors first
Ignore TCP Timestamps in summary
✓ Do not call subdissectors for error packets
✓ TCP Experimental Options with a Magic Number
Disable TCP…
Profile: Default

**Step 10**: Now let's see how a few of these settings affect the packet displays. Click on **frame 8** in ***http-pcaprnet101.pcapng***. Expand the **Transmission Control Protocol line**, the **SEQ/ACK analysis**, and **Timestamps** section in the Packet Details pane.

We can see that Wireshark is not validating the TCP checksum and that 287 bytes of data have been sent, but not acknowledged. In addition, we can see that this frame arrived about 20 milliseconds (0.020 seconds) after the first frame of the TCP conversation (also referred to as the TCP stream) and 778 microseconds (0.000778 seconds) after the previous frame of this TCP conversation.



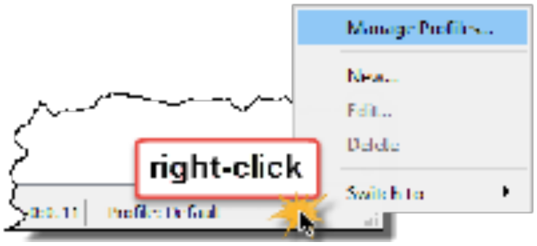
```
8   0.071372   24.6.173.220    209.133.32.69   HTTP   341  GET / HTTP/1.1
9   0.088468   209.133.32.69   24.6.173.220    TCP     60  80 → 21213 [ACK] Seq=1 Ack=288 Wi…

Transmission Control Protocol, Src Port: 21213 (21213), Dst Port: 80 (80), Seq: 1, Ack: 1,…
    Source Port: 21213
    Destination Port: 80
    [Stream index: 0]
    [TCP Segment Len: 287]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 288    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    Header Length: 20 bytes
  > Flags: 0x018 (PSH, ACK)
    Window size value: 16425
    [Calculated window size: 65700]
    [Window size scaling factor: 4]
  > Checksum: 0xb8e6 [validation disabled]
    Urgent pointer: 0
  ⌄ [SEQ/ACK analysis]
      [iRTT: 0.019281000 seconds]
      [Bytes in flight: 287]
  ⌄ [Timestamps]
      [Time since first frame in this TCP stream: 0.020056000 seconds]
      [Time since previous frame in this TCP stream: 0.000778000 seconds]
```

You can easily use the right-click method to change protocol preferences, such as tracking time in each TCP conversation and the number of unacknowledged bytes in a conversation. There are many other application and protocol preference settings that can be set in either the Preferences window or through the right-click method.

## Lab 6: Create a New Profile Based on the *Default* Profile

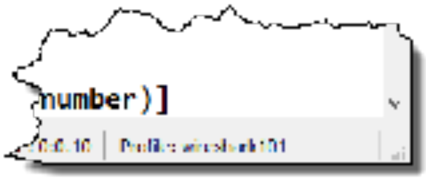
Profiles enable you to work with customized settings to be more efficient when analyzing traffic. In this lab you will create a new profile called "*wireshark101*." You will base it on your *Default* profile to ensure any previously created settings will be copied over to your new profile.

**Step 1**: Right-click on the **Profile** column in the Status Bar and select **Manage Profiles**. (It does not matter what profile is currently listed in the Profile column.)



**Step 2**: Select ***Default*** from the list of available profiles and click the **Copy** button. Type the name ***wireshark101*** and press **Enter**. Click **OK**.

Wireshark now displays your new profile in the Status Bar.



In Lab 6 we worked with some key preference settings (such as *Track number of bytes in flight* and *Calculate conversation timestamps*) in the *Default* profile. Since your new profile is based on the *Default* profile, these preference settings are also set in your *wireshark101* profile.

Wireshark remembers the last profile used when it is restarted. To change to another profile, click on the **Profile** area of the Status Bar and select another profile.

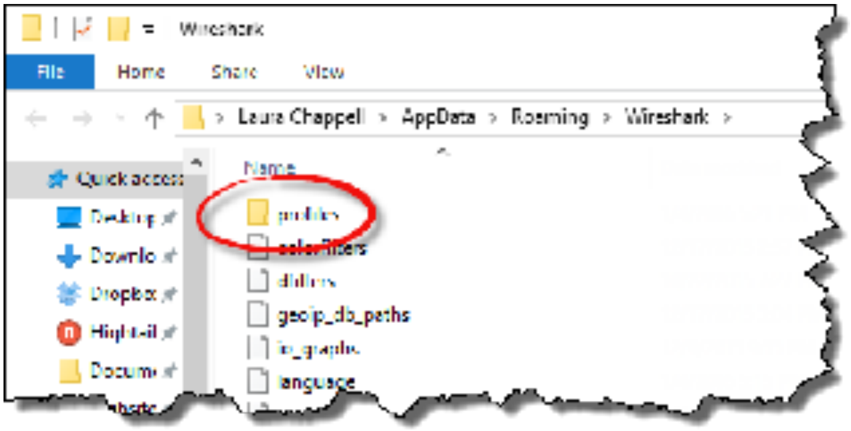
## Lab 7: Import a DNS/HTTP Errors Profile

Once you've created that fabulous profile that detects various types of HTTP or DNS problems perhaps, consider installing that profile on your other Wireshark systems. Since Wireshark bases profiles on text files, this is a simple process.

**Step 1**: The sample profile (***httpdnsprofile101v2.zip***) is located in the Supplements directory of your Student USB stick. This new profile's directory and contents are zipped into a single file.

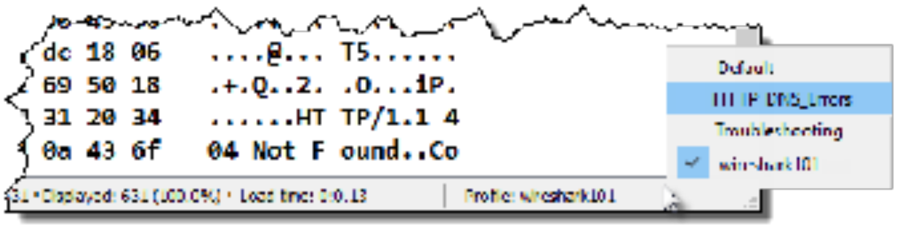
**Step 2**: Select **Help | About Wireshark | Folders**. Double-click on your personal configuration folder to examine the directory structure.

**Step 3**: As mentioned earlier, Wireshark creates a *profiles* directory when you build your first custom profile (as you did in Lab 7). If you do not see a *profiles* directory at this point, you can manually create one or return to and complete Lab 7. Open the ***profiles*** directory.



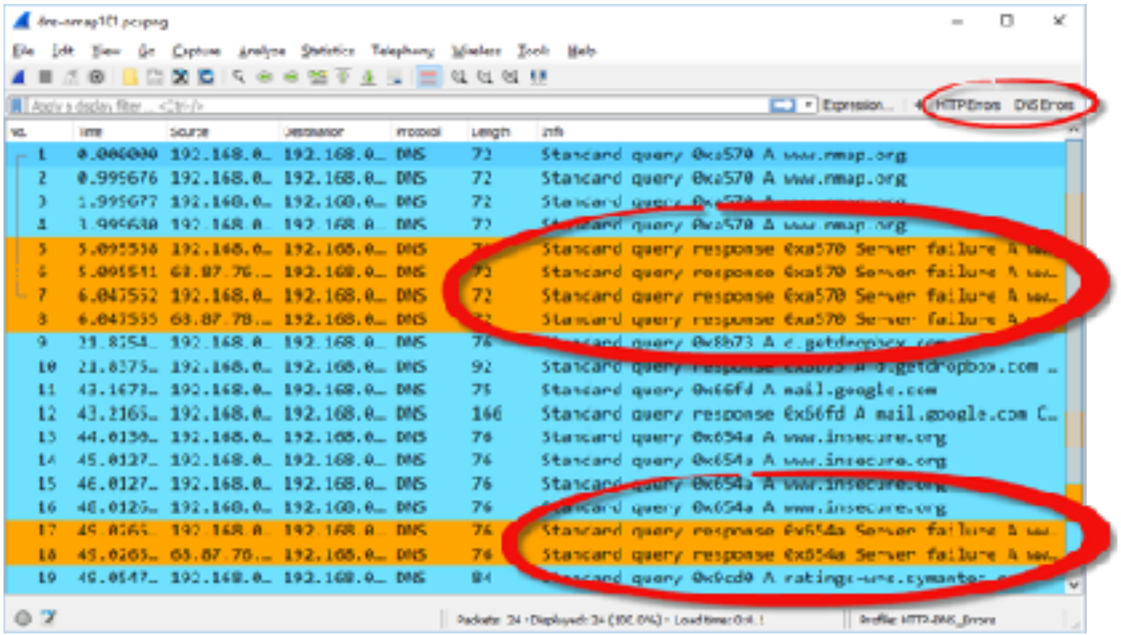
**Step 4**: Extract the ***httpdnsprofile101v2.zip*** file contents into this *profiles* directory. You should see a new directory called *HTTP-DNS_Errors*. Look inside this new directory to see the Wireshark configuration files included in this profile.

**Step 5**: Return to Wireshark and click on the **Profile** column on the Status Bar. You should see the new profile listed. Click on the ***HTTP-DNS_Errors*** profile to examine this new profile.



```
dc 18 06   ....@... T5......
69 50 18   .+.Q..2. .O...1P.
31 20 34   .......HT TP/1.1 4
0a 43 6f   04 Not F ound..Co
```

**Step 6**: Open ***dns-nmap101.pcapng*** while working in your *HTTP-DNS_Errors* profile. You should see some interesting colors in the trace file and two new buttons in the display filter area.



```
No.  Time      Source       Destination  Protocol Length Info
1    0.000000  192.168.0…   192.168.0…   DNS      72     Standard query 0xa570 A www.nmap.org
2    0.999676  192.168.0…   192.168.0…   DNS      72     Standard query 0xa570 A www.nmap.org
3    1.999677  192.168.0…   192.168.0…   DNS      72     Standard query 0xa570 A www.nmap.org
4    3.999680  192.168.0…   192.168.0…   DNS      72     Standard query 0xa570 A www.nmap.org
5    5.095536  192.168.0…   192.168.0…   DNS      72     Standard query response 0xa570 Server failure A ww…
6    5.095541  68.87.76.…   192.168.0…   DNS      72     Standard query response 0xa570 Server failure A ww…
7    6.047552  192.168.0…   192.168.0…   DNS      72     Standard query response 0xa570 Server failure A ww…
8    6.047555  68.87.78.…   192.168.0…   DNS      72     Standard query response 0xa570 Server failure A ww…
9    21.8754…  192.168.0…   192.168.0…   DNS      76     Standard query 0x8b73 A d.getdropbox.com
10   21.8375…  192.168.0…   192.168.0…   DNS      92     Standard query response 0x8b73 A d.getdropbox.com …
11   43.1673…  192.168.0…   192.168.0…   DNS      75     Standard query 0x66fd A mail.google.com
12   43.2165…  192.168.0…   192.168.0…   DNS      166    Standard query response 0x66fd A mail.google.com C…
13   44.0130…  192.168.0…   192.168.0…   DNS      76     Standard query 0x654a A www.insecure.org
14   45.0127…  192.168.0…   192.168.0…   DNS      76     Standard query 0x654a A www.insecure.org
15   46.0127…  192.168.0…   192.168.0…   DNS      76     Standard query 0x654a A www.insecure.org
16   48.0125…  192.168.0…   192.168.0…   DNS      76     Standard query 0x654a A www.insecure.org
17   49.0165…  192.168.0…   192.168.0…   DNS      76     Standard query response 0x654a Server failure A ww…
18   49.0165…  68.87.76.…   192.168.0…   DNS      76     Standard query response 0x654a Server failure A ww…
19   49.0547…  192.168.0…   192.168.0…   DNS      84     Standard query 0x9cd0 A ratings-wrs.symantec…
```

```
Packets: 24 · Displayed: 24 (100.0%) · Load time: 0:0.1        Profile: HTTP-DNS_Errors
```

**Step 7**: Lab Clean-up Click the **Profile** column on the Status Bar and select your ***wireshark101*** profile. You will continue to enhance the *wireshark101* profile in upcoming Sections of this course.

Remember that profiles are simply a collection of configuration text files. It is easy to move single elements of a profile or entire profiles to other machines. If you work with a troubleshooting team, consider creating common Wireshark profiles that the entire team can use.

**TIP**

**Some configuration text files, such as the *recent* configuration file, contain directory paths. This may generate Wireshark startup errors when you move these types of configuration files to another system that does not have the same directory paths in place. You could either avoid moving these files to another system or edit the relevant configuration files to match the directory structure of the target system.**

## Lab 8: Spot Path and Server Latency Problems

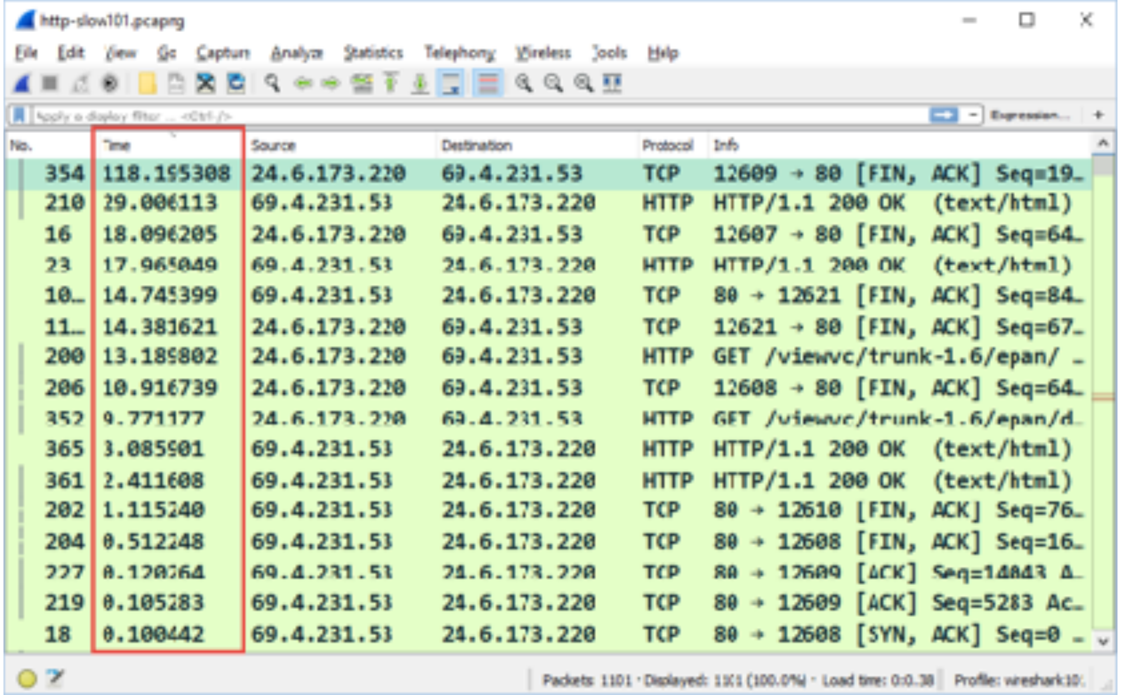
Let's practice using these two columns to detect latency. In this lab you will set the **Time** column to *Seconds Since Previous Displayed Packet* and add the **TCP Delta** column.

You may have some of these columns set already if you followed along with the previous section.

**Step 1:** Open ***http-slow101.pcapng***.

**Step 2:** Right-click the **Length** column heading unselect the **Length** column to hide it. This provides more room for your new column.

**Step 3:** Select **View | Time Display Format | Seconds Since Previous Displayed Packet**. Click on your **Time** column heading twice to sort from high to low. Click the **First Packet** button on the main toolbar. We can see some very high delays in this trace file.


http-slow101.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...

| No. | Time | Source | Destination | Protocol | Info |
|---|---|---|---|---|---|
| 354 | 118.155308 | 24.6.173.220 | 69.4.231.53 | TCP | 12609 → 80 [FIN, ACK] Seq=19… |
| 210 | 29.006113 | 69.4.231.53 | 24.6.173.220 | HTTP | HTTP/1.1 200 OK (text/html) |
| 16 | 18.096205 | 24.6.173.220 | 69.4.231.53 | TCP | 12607 → 80 [FIN, ACK] Seq=64… |
| 23 | 17.965049 | 69.4.231.53 | 24.6.173.220 | HTTP | HTTP/1.1 200 OK (text/html) |
| 10… | 14.745399 | 69.4.231.53 | 24.6.173.220 | TCP | 80 → 12621 [FIN, ACK] Seq=84… |
| 11… | 14.381621 | 24.6.173.220 | 69.4.231.53 | TCP | 12621 → 80 [FIN, ACK] Seq=67… |
| 200 | 13.189802 | 24.6.173.220 | 69.4.231.53 | HTTP | GET /viewvc/trunk-1.6/epan/ … |
| 206 | 10.916739 | 24.6.173.220 | 69.4.231.53 | TCP | 12608 → 80 [FIN, ACK] Seq=64… |
| 352 | 9.771177 | 24.6.173.220 | 69.4.231.53 | HTTP | GET /viewvc/trunk-1.6/epan/d… |
| 365 | 3.085901 | 69.4.231.53 | 24.6.173.220 | HTTP | HTTP/1.1 200 OK (text/html) |
| 361 | 2.411608 | 69.4.231.53 | 24.6.173.220 | HTTP | HTTP/1.1 200 OK (text/html) |
| 202 | 1.115240 | 69.4.231.53 | 24.6.173.220 | TCP | 80 → 12610 [FIN, ACK] Seq=76… |
| 204 | 0.512248 | 69.4.231.53 | 24.6.173.220 | TCP | 80 → 12608 [FIN, ACK] Seq=16… |
| 227 | 0.120264 | 69.4.231.53 | 24.6.173.220 | TCP | 80 → 12609 [ACK] Seq=14843 A… |
| 219 | 0.105283 | 69.4.231.53 | 24.6.173.220 | TCP | 80 → 12609 [ACK] Seq=5283 Ac… |
| 18 | 0.100442 | 69.4.231.53 | 24.6.173.220 | TCP | 80 → 12608 [SYN, ACK] Seq=0 … |

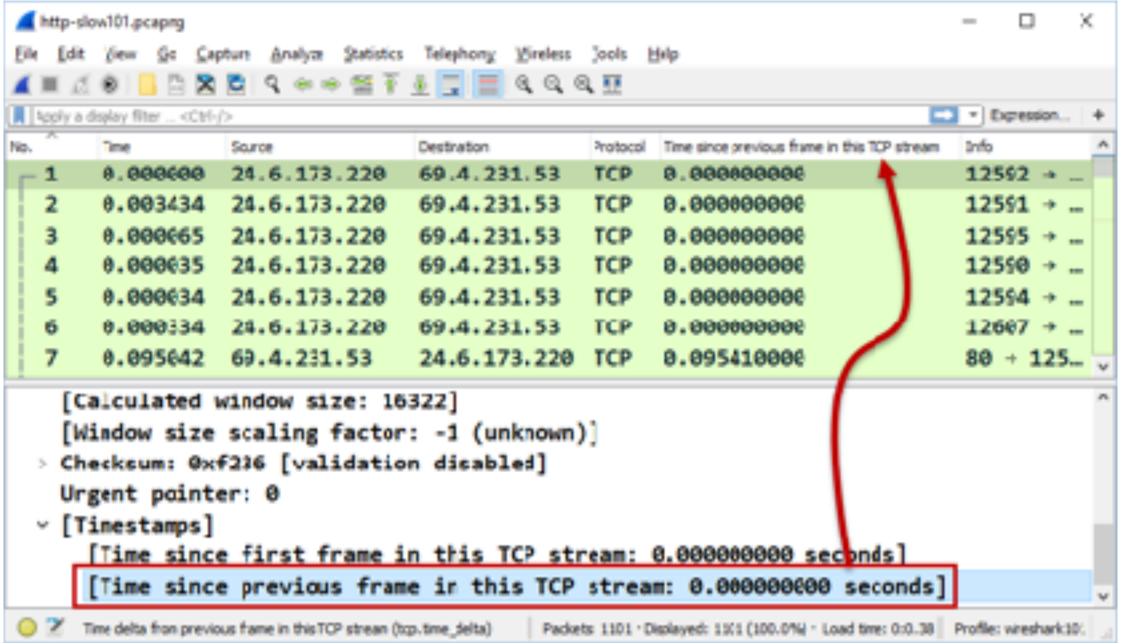
Packets: 1101 · Displayed: 1101 (100.0%) · Load time: 0:0.38 Profile: wireshark10…

Now let's see what happens when we add and work with a column that depicts TCP conversation timestamps.

**Step 4**: Click on the **No.** (Number) column heading to return the trace file to its default sort order.

Scroll up or click the **Go to First** button on the main toolbar to go to **frame 1**.

**Step 5**: Right-click the **TCP header** in the Packet Details pane of frame 1 and select **Expand Subtrees**. Scroll down and right-click on the **Time since previous frame in this TCP stream** and select **Apply as Column**. You now have a new column in the Packet List pane, as shown below.

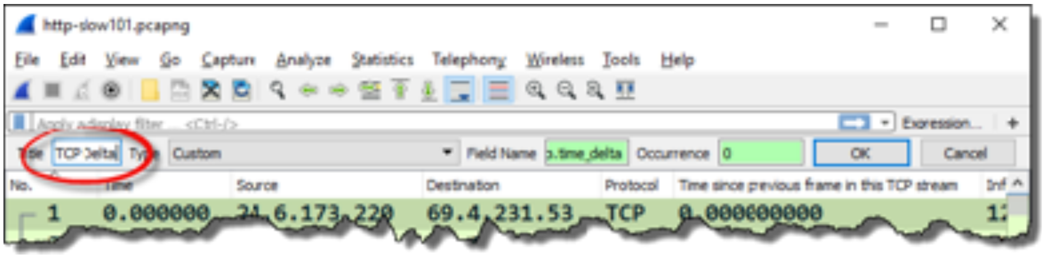

http-slow101.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
+

```
No.  Time      Source        Destination   Protocol Time since previous frame in this TCP stream Info
1    0.000000  24.6.173.220  69.4.231.53   TCP      0.000000000                                  12592 → ...
2    0.003434  24.6.173.220  69.4.231.53   TCP      0.000000000                                  12591 → ...
3    0.000665  24.6.173.220  69.4.231.53   TCP      0.000000000                                  12595 → ...
4    0.000035  24.6.173.220  69.4.231.53   TCP      0.000000000                                  12590 → ...
5    0.000034  24.6.173.220  69.4.231.53   TCP      0.000000000                                  12594 → ...
6    0.000334  24.6.173.220  69.4.231.53   TCP      0.000000000                                  12607 → ...
7    0.095642  69.4.231.53   24.6.173.220  TCP      0.095410000                                  80 → 125...
```

```
    [Calculated window size: 16322]
    [Window size scaling factor: -1 (unknown)]
  > Checksum: 0xf236 [validation disabled]
    Urgent pointer: 0
  v [Timestamps]
      [Time since first frame in this TCP stream: 0.000000000 seconds]
      [Time since previous frame in this TCP stream: 0.000000000 seconds]
```

Time delta from previous frame in this TCP stream (tcp.time_delta) | Packets: 1101 · Displayed: 1101 (100.0%) · Load time: 0:0.38 | Profile: wireshark101

**Step 6**: Right-click on the new column and select **Edit Column**. Type **TCP Delta** in the Title area and click **OK**.


http-slow101.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
+
OK
Cancel

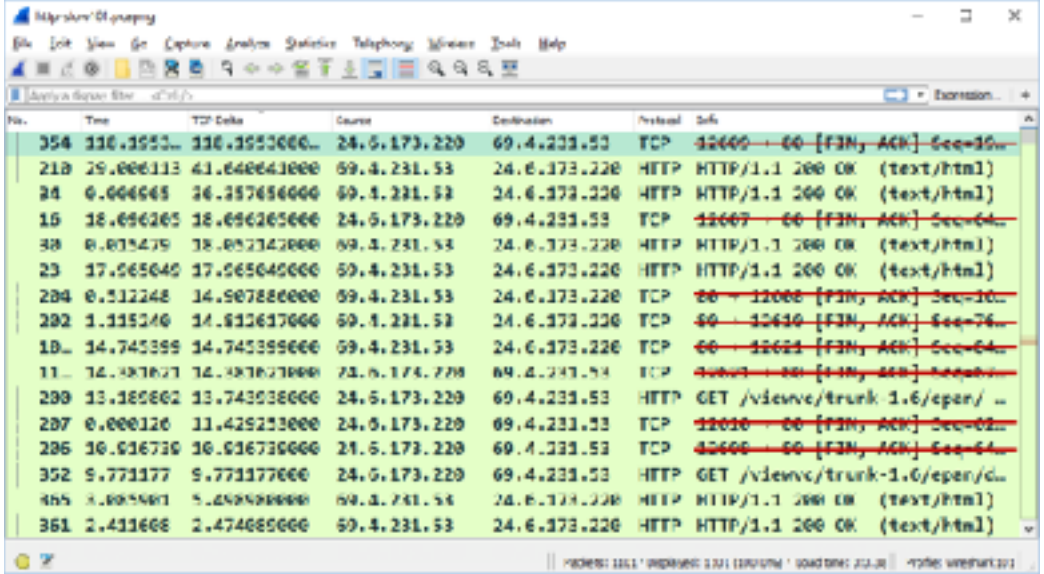
Title: TCP Delta Type: Custom Field Name: tcp.time_delta Occurrence: 0

```
No.  Time      Source        Destination   Protocol Time since previous frame in this TCP stream Info
1    0.000000  24.6.173.220  69.4.231.53   TCP      0.000000000                                  12
```

As we sort on the **TCP Delta** column, keep in mind the types of traffic that can contain "normal delays" as listed in *Don't Get Fooled – Some Delays are Normal* on page 88.

**Step 7:** Click on your new **TCP Delta** column heading and drag the column to the right of the existing **Time** column. Click twice on your new **TCP Delta** column heading to sort from high to low. Since there are multiple TCP conversations intertwined in this trace file, this **TCP Delta** column gives an accurate display of latency times in the trace file.

In the image below, we scrolled to the right to view more of the Info column (our Time column is no longer in view).



| No. | Time | TCP Delta | Source | Destination | Protocol | Info |
|---|---|---|---|---|---|---|
| 354 | 116.1553… | 116.1553000… | 24.6.173.220 | 69.4.231.53 | TCP | ~~12660 → 80 [FIN, ACK] Seq=16…~~ |
| 218 | 29.006113 | 41.646641000 | 69.4.231.53 | 24.6.173.220 | HTTP | HTTP/1.1 200 OK (text/html) |
| 24 | 0.666665 | 36.357656000 | 69.4.231.53 | 24.6.173.220 | HTTP | HTTP/1.1 200 OK (text/html) |
| 16 | 18.696265 | 18.696265000 | 24.6.173.220 | 69.4.231.53 | TCP | ~~12667 → 80 [FIN, ACK] Seq=64…~~ |
| 38 | 0.015479 | 18.057142000 | 69.4.231.53 | 24.6.173.220 | HTTP | HTTP/1.1 200 OK (text/html) |
| 23 | 17.565646 | 17.565646000 | 69.4.231.53 | 24.6.173.220 | HTTP | HTTP/1.1 200 OK (text/html) |
| 204 | 0.512248 | 14.907888000 | 69.4.231.53 | 24.6.173.220 | TCP | ~~80 → 12668 [FIN, ACK] Seq=10…~~ |
| 202 | 1.115346 | 14.613617000 | 69.4.231.53 | 24.6.173.220 | TCP | ~~80 → 12610 [FIN, ACK] Seq=76…~~ |
| 10… | 14.745399 | 14.745399000 | 69.4.231.53 | 24.6.173.220 | TCP | ~~80 → 12621 [FIN, ACK] Seq=64…~~ |
| 11… | [illegible] | [illegible] | [illegible] | [illegible] | TCP | [illegible] |
| 200 | 13.185862 | 13.743038000 | 24.6.173.220 | 69.4.231.53 | HTTP | GET /viewvc/trunk-1.6/epan/ … |
| 207 | 0.000126 | 11.429253000 | 24.6.173.220 | 69.4.231.53 | TCP | ~~12616 → 80 [FIN, ACK] Seq=62…~~ |
| 206 | 10.916739 | 10.916739000 | 24.6.173.220 | 69.4.231.53 | TCP | ~~12666 → 80 [FIN, ACK] Seq=64…~~ |
| 352 | 9.771177 | 9.771177000 | 24.6.173.220 | 69.4.231.53 | HTTP | GET /viewvc/trunk-1.6/epan/d… |
| 365 | [illegible] | [illegible] | 69.4.231.53 | 24.6.173.220 | HTTP | HTTP/1.1 200 OK (text/html) |
| 361 | 2.411608 | 2.474085000 | 69.4.231.53 | 24.6.173.220 | HTTP | HTTP/1.1 200 OK (text/html) |

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression… +

Do you see anything in common with the top delays in the traffic? There are several very large delays before the HTTP server said “OK.” You can probably imagine that the user would complain about terrible performance when browsing to this web site.

**Step 8**: Lab Clean-up Click once on the **No.** (Number) column heading to sort from low to high. This is the original sorting order of trace files.

Right-click on the **TCP Delta** column heading and unselect that column from the list to hide it. If you want to view this column again later, you can right-click on any column heading and select it from the column list.

Look at the TCP delta times in your web browsing sessions, network logins, or email traffic. Get a feel for the round trip latency times from your client to numerous hosts.