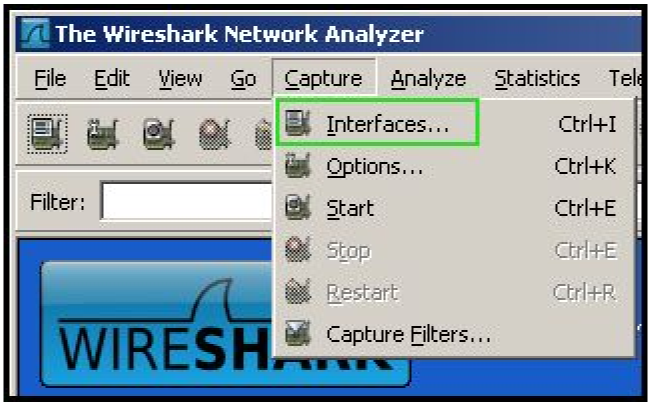
Packet Analysis Using Wireshark (GW)

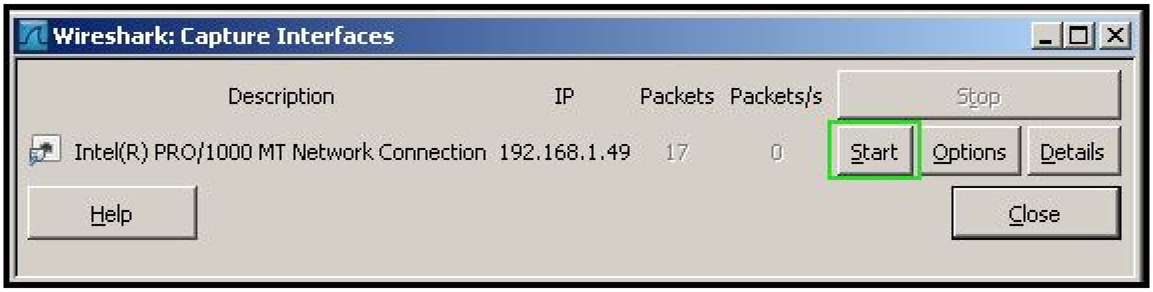
A. A Brief Introduction to Wireshark

Wireshark is a protocol analyzer, or "packet sniffer" application, used for network troubleshooting, analysis, software and protocol development, and education. It allows the user to see all network traffic being passed over the network by putting the network card into the promiscuous mode. In this activity, you will use Wireshark on Windows 7 to analyze packets between your and your partner's computers.

1. Double-click the icon for **Wireshark** on the Desktop to open Wireshark.
2. In the menu bar, select **Capture** and then **Interfaces**.

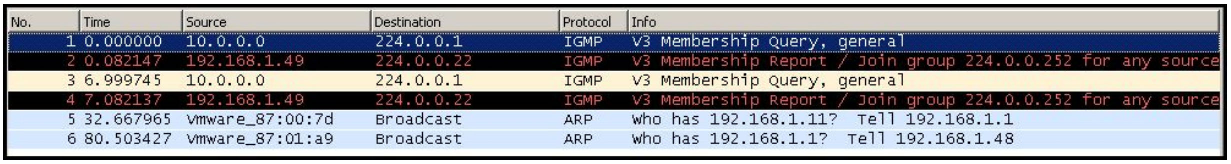


3. In the Capture Interfaces window, you will see the Ethernet adapter of your virtual computer listed. The IP address shown is the IP address of your computer that is dynamically assigned. To begin collecting packets that are being sent to and from this adapter, click **Start**.



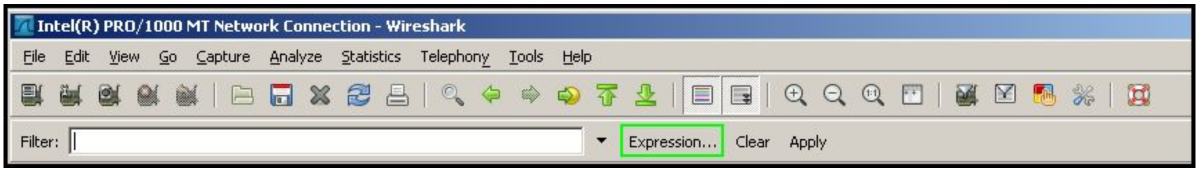
Description	IP	Packets	Packets/s
Intel(R) PRO/1000 MT Network Connection	192.168.1.49	17	0

4. As soon as you click **Start**, Wireshark will begin capturing packets. You will see the packets appear as a list in the top section of the Wireshark window.

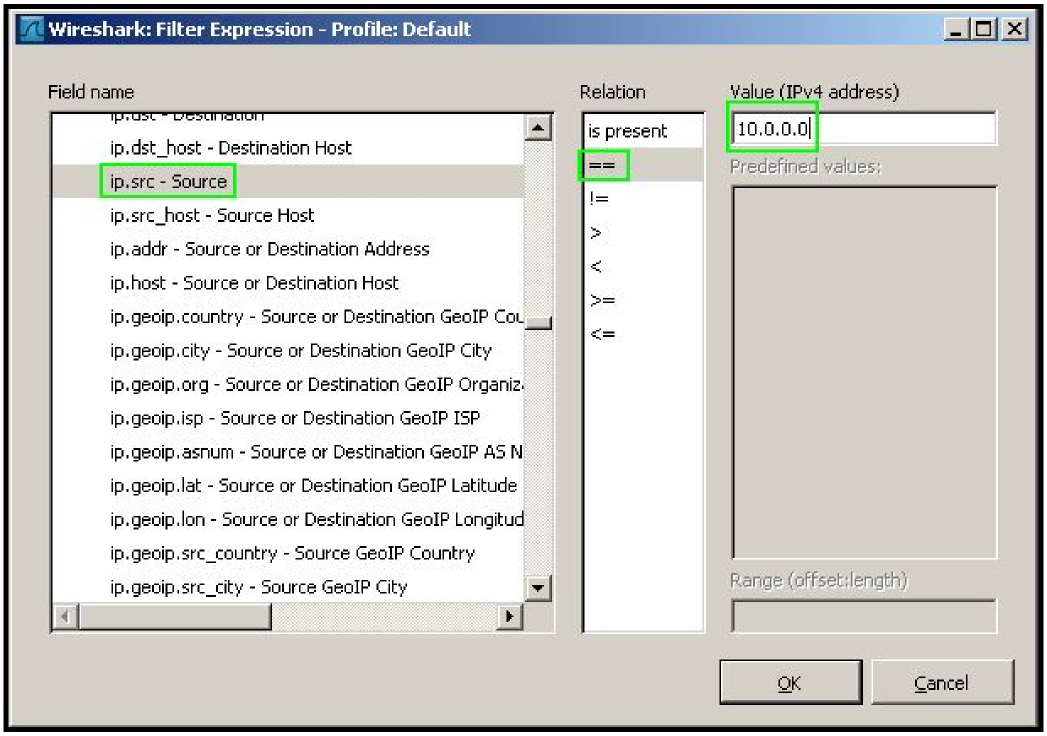


No.	Time	Source	Destination	Protocol	Info
1	0.000000	10.0.0.0	224.0.0.1	IGMP	V3 Membership Query, general
2	0.082147	192.168.1.49	224.0.0.22	IGMP	V3 Membership Report / Join group 224.0.0.252 for any source
3	6.999745	10.0.0.0	224.0.0.1	IGMP	V3 Membership Query, general
4	7.082137	192.168.1.49	224.0.0.22	IGMP	V3 Membership Report / Join group 224.0.0.252 for any source
5	32.667965	Vmware_87:00:7d	Broadcast	ARP	Who has 192.168.1.11? Tell 192.168.1.1
6	80.503427	Vmware_87:01:a9	Broadcast	ARP	Who has 192.168.1.1? Tell 192.168.1.48

5. Open Command Prompt at the Start Menu, Click yes if you are prompt.
6. Type `ping 10.0.0.1` in the command prompt to create some network traffic, and then close the command prompt window.
7. To stop capturing packets, go to **Capture** and click **Stop**.
8. Note that you can filter the packets by entering a **filter requirement**. Click the **Expression** button below the menu bar.



9. Locate **IP - Internet Protocol** in the Field name list. Expand it, then select **ip.src - Source**. Select the **==** relation and enter **10.0.0.0** as the Value (IPv4 address), then click OK.

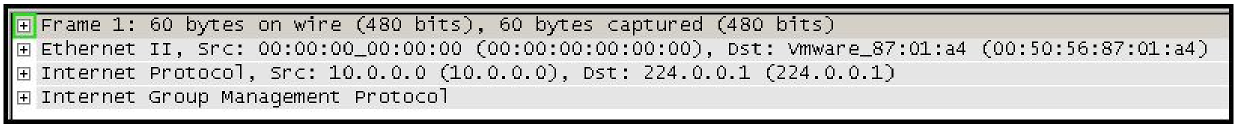

Wireshark: Filter Expression - Profile: Default
Field name
ip.dst - Destination
ip.dst_host - Destination Host
ip.src - Source
ip.src_host - Source Host
ip.addr - Source or Destination Address
ip.host - Source or Destination Host
ip.geoip.country - Source or Destination GeoIP Cou
ip.geoip.city - Source or Destination GeoIP City
ip.geoip.org - Source or Destination GeoIP Organiz
ip.geoip.isp - Source or Destination GeoIP ISP
ip.geoip.asnum - Source or Destination GeoIP AS N
ip.geoip.lat - Source or Destination GeoIP Latitude
ip.geoip.lon - Source or Destination GeoIP Longitud
ip.geoip.src_country - Source GeoIP Country
ip.geoip.src_city - Source GeoIP City
Relation
is present
==
!=
>
<
>=
<=
Value (IPv4 address)
Predefined values:
Range (offset:length)
OK
Cancel

Value (IPv4 address): 10.0.0.0

10. You will now see your filter requirement highlighted in green. Click the **Apply** button

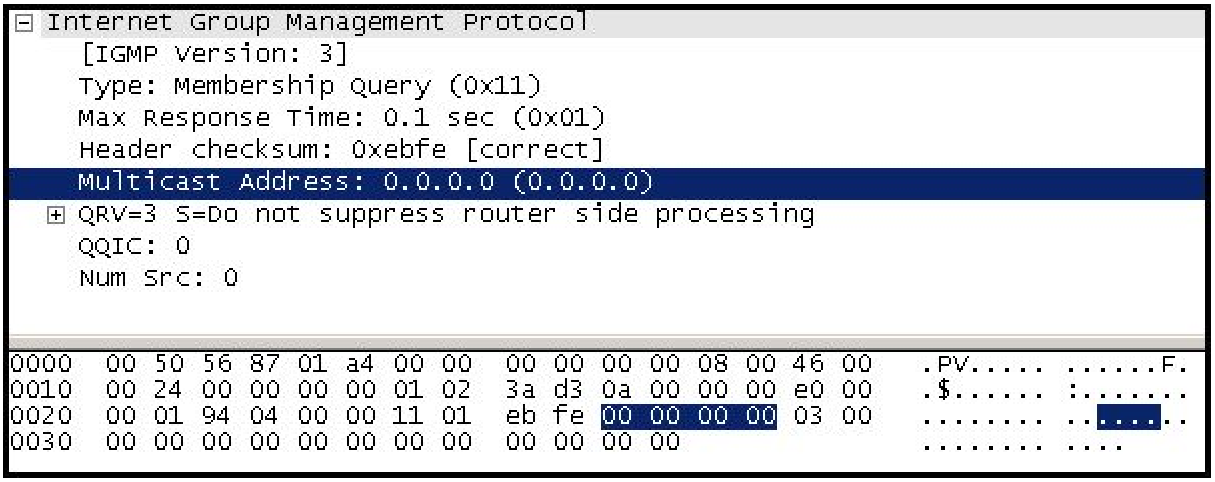
below the toolbar to apply the filter. After doing so, you should see only packets with a source IP address of 10.0.0.0 appear in the packet list. Filtering makes it easier to see particular packets you are looking for.

11. Click on a packet in the list. Notice the content shown in the middle section of the Wireshark window. Each category can be expanded by clicking the “**+**” symbol. Each high level "+" represents a packet header. **What are the headers of a ping packet?**



```
⊞ Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
⊞ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: Vmware_87:01:a4 (00:50:56:87:01:a4)
⊞ Internet Protocol, Src: 10.0.0.0 (10.0.0.0), Dst: 224.0.0.1 (224.0.0.1)
⊞ Internet Group Management Protocol
```

12. Expand the categories and look through the information found for the packet. As you can see, Wireshark is a powerful tool.
13. Notice that when you click on a part of the packet, the relevant portion is highlighted in the bottom section of the Wireshark window. The bottom section contains the actual data in its original form (lefthand side) and hexadecimal format (right hand side).



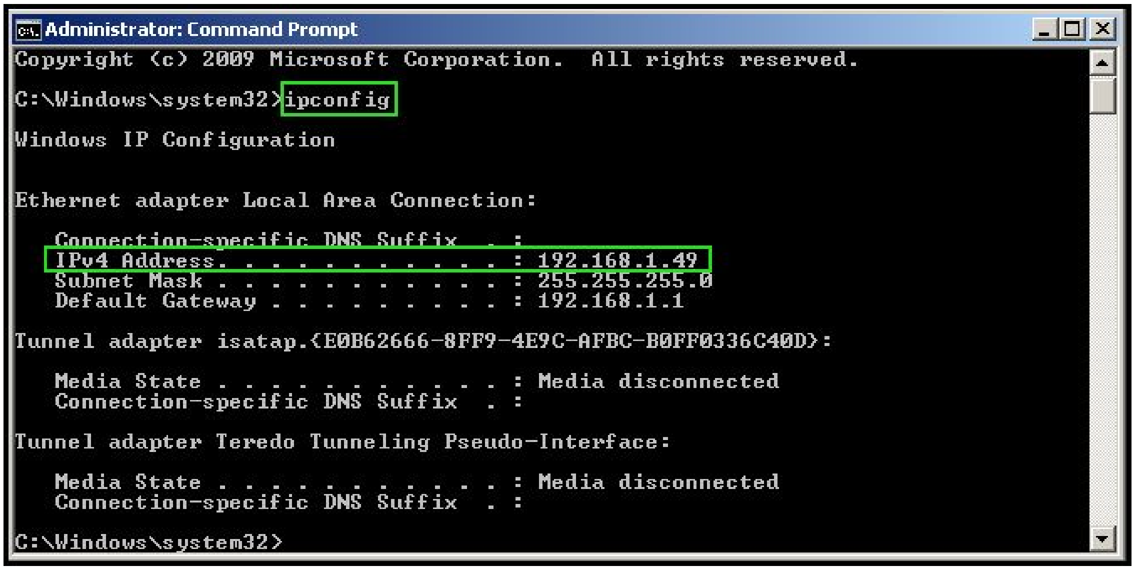
```
⊟ Internet Group Management Protocol
    [IGMP Version: 3]
    Type: Membership Query (0x11)
    Max Response Time: 0.1 sec (0x01)
    Header checksum: 0xebfe [correct]
    Multicast Address: 0.0.0.0 (0.0.0.0)
  ⊞ QRV=3 S=Do not suppress router side processing
    QQIC: 0
    Num Src: 0

0000  00 50 56 87 01 a4 00 00  00 00 00 00 08 00 46 00   .PV..... ......F.
0010  00 24 00 00 00 00 01 02  3a d3 0a 00 00 00 e0 00   .$...... :.......
0020  00 01 94 04 00 00 11 01  eb fe 00 00 00 00 03 00   ........ ........
0030  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....
```

B. Testing the IIS Web Server

Internet Information Services (IIS) is Microsoft’s **Web Server** Application that makes it easy to publish information on the Internet. A web server is a computer that is responsible for accepting HyperText Transport Protocol (HTTP) requests from Web browsers (clients) and serving them Web pages, which are usually HyperText Markup Language (HTML) documents. We will use the IIS web server to generate packets to analyze with Wireshark in this activity.

IIS is already installed and running in your Windows 7 computer. Before starting, test whether you can access your teammate’s default website. To do this, you need to know your teammate’s IP address.

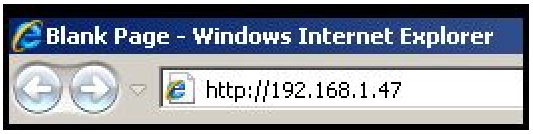
1. Click on **Start**, go to **All Programs**, click **Accessories**, and select **Command Prompt**. If you receive a pop-up, click **Yes**.
2. In the command prompt, type **`ipconfig`** and press Enter. This command will fetch your computer’s IP address. Under **Ethernet adapter Local Area Connection**, look for **IPv4 Address** and make note of what your IP address is and share this with your partner.



```
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.1.49
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1

Tunnel adapter isatap.{E0B62666-8FF9-4E9C-AFBC-B0FF0336C40D}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter Teredo Tunneling Pseudo-Interface:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Windows\system32>
```

Your IP address	**Your teammate’s IP address**

3. Now, you can attempt to access your teammate’s default website. Begin by opening **Internet Explorer**.
4. In the URL bar, enter **`http://yourpartner’sIPaddress.`** “yourpartner’sIPaddress” is the IP address your partner found in the previous step. An example is shown below.



http://192.168.1.47

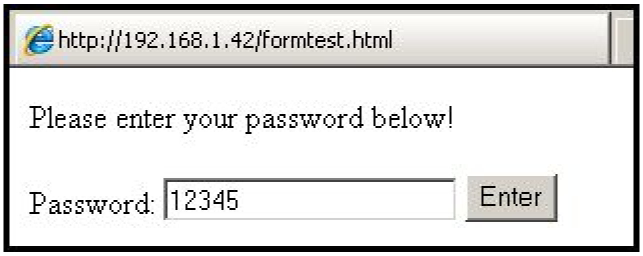
5. If the page is able to load, then your teammate’s website is working.

6. Now, test your web server by typing **`http://localhost`** into the URL bar of Internet Explorer.
7. Close Internet Explorer.

C. Using Wireshark to Capture HTTP Packets

In this exercise, you will use Wireshark to capture packets. Please coordinate with your teammate. One of you will be PCa and the other will be PCb in the following instructions. PCa will connect to PCb's web site and PCb will capture packets using Wireshark.

1. In **PCb**, make sure that Wireshark is closed. Them, follow **steps 1-4** in **Section A** to open Wireshark and begin capturing packets on your network interface.
2. In **PCa**, open Internet Explorer and type **`http://yourpartner'sIPaddress/formtest.html`** where "yourpartner'sIPaddress" is your partner's actual IP address found previously.
3. Enter a password of **`12345`** into the password box, then click **Enter**.

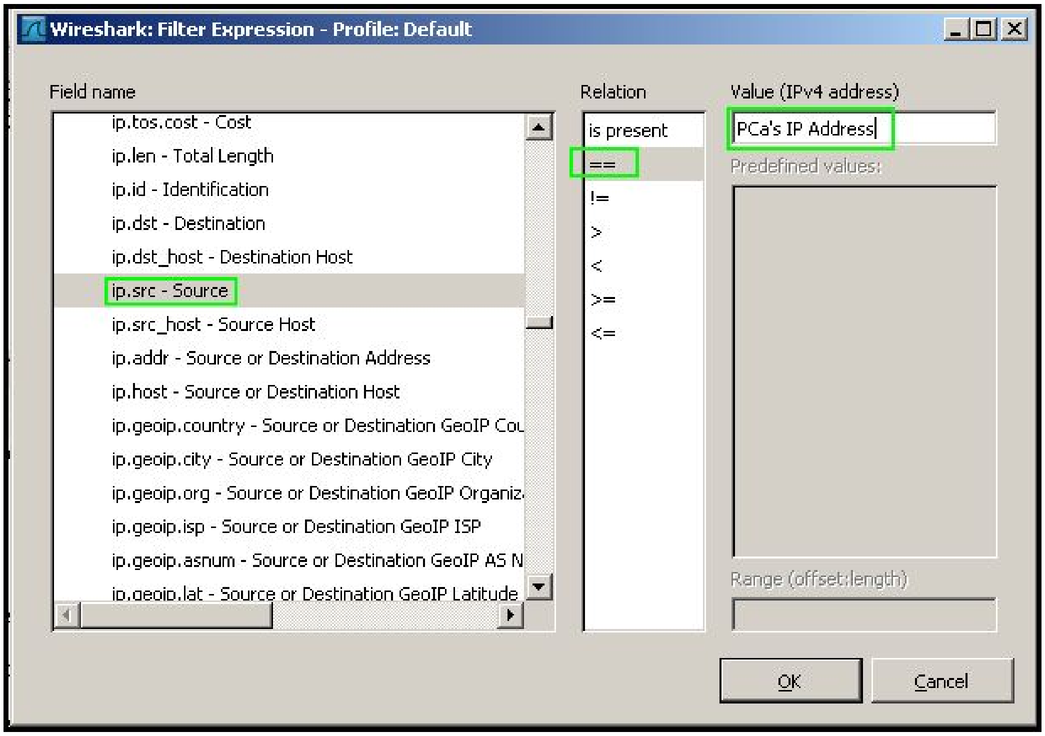


http://192.168.1.42/formtest.html

Please enter your password below!

Password: 12345

4. In **PCb**, you may stop capturing packets by clicking **Capture** and selecting **Stop**. If you still have the filter from a previous step applied, remove it by clicking **Clear** next to the filter textbox below the toolbar.
5. Look through the variety of packets captured by Wireshark. Because we are interested in the packets generated by PCa, we will add a filter to help find the valuable packets. Click the **Expression** button below the toolbar, then find **IP - Internet Protocol** in the Field name list. Expand it, then select **ip.src - Source**. Select the **==** relation and enter **PCa's IP address** as the Value (IPv4 address), then click OK.



Wireshark: Filter Expression - Profile: Default

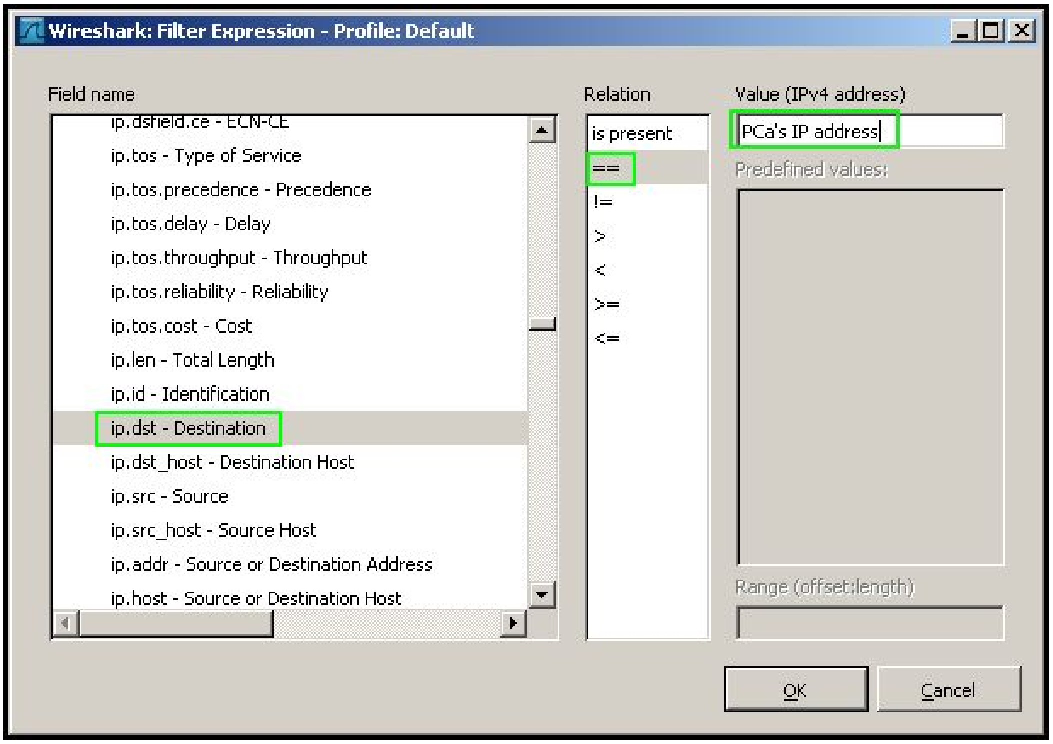
Field name: ip.src - Source

Relation: ==

Value (IPv4 address): PCa's IP Address

ip.tos.cost - Cost
ip.len - Total Length
ip.id - Identification
ip.dst - Destination
ip.dst_host - Destination Host
ip.src_host - Source Host
ip.addr - Source or Destination Address
ip.host - Source or Destination Host
ip.geoip.country - Source or Destination GeoIP Cou
ip.geoip.city - Source or Destination GeoIP City
ip.geoip.org - Source or Destination GeoIP Organiz
ip.geoip.isp - Source or Destination GeoIP ISP
ip.geoip.asnum - Source or Destination GeoIP AS N
ip.geoip.lat - Source or Destination GeoIP Latitude
is present
!=
>
<
>=
<=
Predefined values:
Range (offset:length)
OK
Cancel

6. Click **Expression** once more, select **IP - Internet Protocol**, and then click **ip.dst - Destination**. Select the **==** operator and enter PCa's IP address, then click OK.



Wireshark: Filter Expression - Profile: Default

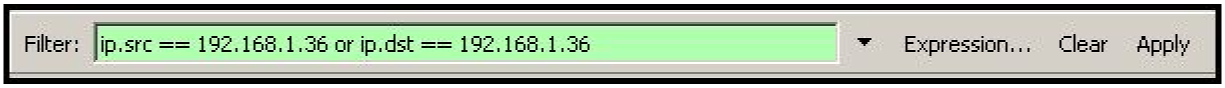
Field name: ip.dst - Destination

Relation: ==

Value (IPv4 address): PCa's IP address

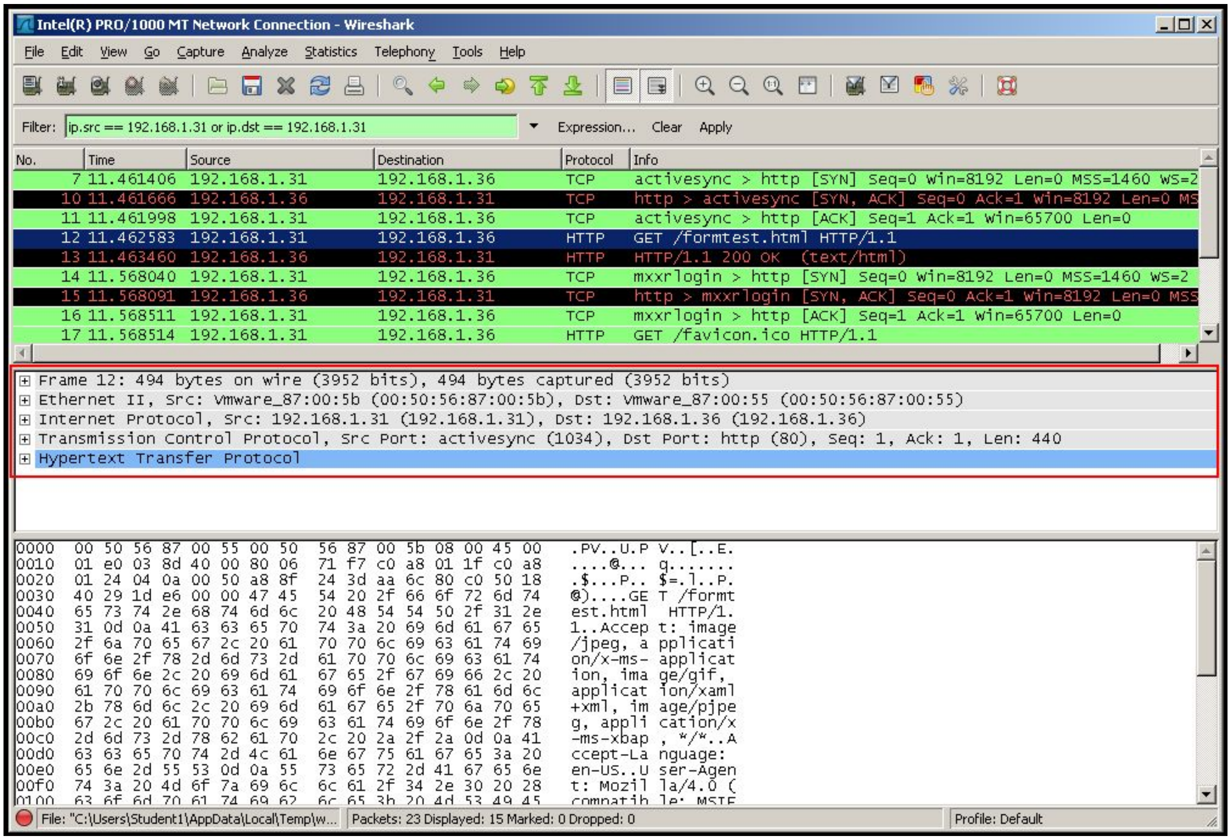
ip.dsfield.ce - ECN-CE
ip.tos - Type of Service
ip.tos.precedence - Precedence
ip.tos.delay - Delay
ip.tos.throughput - Throughput
ip.tos.reliability - Reliability
ip.tos.cost - Cost
ip.len - Total Length
ip.id - Identification
ip.dst_host - Destination Host
ip.src - Source
ip.src_host - Source Host
ip.addr - Source or Destination Address
ip.host - Source or Destination Host
is present
!=
>
<
>=
<=
Predefined values:
Range (offset:length)
OK
Cancel

7. You will now see your filter requirement highlighted in green. Because we want to show the packets with PCa as the source, as well as packets with PCa as the destination, we need to add **or** between the two filter expressions. Type **`or`** between the two statements as shown below.



Filter: ip.src == 192.168.1.36 or ip.dst == 192.168.1.36

8. Click the **Apply** button next to the filter textbox to apply the filter. After doing so, you should only see packets **to and from** your partner's computer appear in the packet list.
9. Try to analyze the content of those packets – make sure to have the middle section of the Wireshark window expanded as seen below. If it isn't expanded, hover the mouse above the bottom section until a double arrow appears, then click and drag upward to reveal the middle section.



Filter: ip.src == 192.168.1.31 or ip.dst == 192.168.1.31

```
No.   Time       Source        Destination   Protocol Info
    7 11.461406  192.168.1.31  192.168.1.36  TCP      activesync > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2
   10 11.461666  192.168.1.36  192.168.1.31  TCP      http > activesync [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MS
   11 11.461998  192.168.1.31  192.168.1.36  TCP      activesync > http [ACK] Seq=1 Ack=1 Win=65700 Len=0
   12 11.462583  192.168.1.31  192.168.1.36  HTTP     GET /formtest.html HTTP/1.1
   13 11.463460  192.168.1.36  192.168.1.31  HTTP     HTTP/1.1 200 OK  (text/html)
   14 11.568040  192.168.1.31  192.168.1.36  TCP      mxxrlogin > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2
   15 11.568091  192.168.1.36  192.168.1.31  TCP      http > mxxrlogin [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS
   16 11.568511  192.168.1.31  192.168.1.36  TCP      mxxrlogin > http [ACK] Seq=1 Ack=1 Win=65700 Len=0
   17 11.568514  192.168.1.31  192.168.1.36  HTTP     GET /favicon.ico HTTP/1.1
```

```
⊞ Frame 12: 494 bytes on wire (3952 bits), 494 bytes captured (3952 bits)
⊞ Ethernet II, Src: Vmware_87:00:5b (00:50:56:87:00:5b), Dst: Vmware_87:00:55 (00:50:56:87:00:55)
⊞ Internet Protocol, Src: 192.168.1.31 (192.168.1.31), Dst: 192.168.1.36 (192.168.1.36)
⊞ Transmission Control Protocol, Src Port: activesync (1034), Dst Port: http (80), Seq: 1, Ack: 1, Len: 440
⊞ Hypertext Transfer Protocol
```

```
0000  00 50 56 87 00 55 00 50  56 87 00 5b 08 00 45 00   .PV..U.P V..[..E.
0010  01 e0 03 8d 40 00 80 06  71 f7 c0 a8 01 1f c0 a8   ....@... q.......
0020  01 24 04 0a 00 50 a8 8f  24 3d aa 6c 80 c0 50 18   .$...P.. $=.l..P.
0030  40 29 1d e6 00 00 47 45  54 20 2f 66 6f 72 6d 74   @)....GE T /formt
0040  65 73 74 2e 68 74 6d 6c  20 48 54 54 50 2f 31 2e   est.html  HTTP/1.
0050  31 0d 0a 41 63 63 65 70  74 3a 20 69 6d 61 67 65   1..Accep t: image
0060  2f 6a 70 65 67 2c 20 61  70 70 6c 69 63 61 74 69   /jpeg, a pplicati
0070  6f 6e 2f 78 2d 6d 73 2d  61 70 70 6c 69 63 61 74   on/x-ms- applicat
0080  69 6f 6e 2c 20 69 6d 61  67 65 2f 67 69 66 2c 20   ion, ima ge/gif, 
0090  61 70 70 6c 69 63 61 74  69 6f 6e 2f 78 61 6d 6c   applicat ion/xaml
00a0  2b 78 6d 6c 2c 20 69 6d  61 67 65 2f 70 6a 70 65   +xml, im age/pjpe
00b0  67 2c 20 61 70 70 6c 69  63 61 74 69 6f 6e 2f 78   g, appli cation/x
00c0  2d 6d 73 2d 78 62 61 70  2c 20 2a 2f 2a 0d 0a 41   -ms-xbap , */*..A
00d0  63 63 65 70 74 2d 4c 61  6e 67 75 61 67 65 3a 20   ccept-La nguage: 
00e0  65 6e 2d 55 53 0d 0a 55  73 65 72 2d 41 67 65 6e   en-US..U ser-Agen
00f0  74 3a 20 4d 6f 7a 69 6c  6c 61 2f 34 2e 30 20 28   t: Mozil la/4.0 (
0100  63 6f 6d 70 61 74 69 62  6c 65 3b 20 4d 53 49 45   compatib le; MSIE
```

File: "C:\Users\Student1\AppData\Local\Temp\w... Packets: 23 Displayed: 15 Marked: 0 Dropped: 0 Profile: Default

10. **Leaving Wireshark open on PCb**, repeat **steps 1-7** of this section but reverse the roles for PCa and PCb (**this time, PCa will capture packets, PCb will load the website, and the packets will be filtered by PCb's IP address**).
11. Once **both** PCa and PCb have packets filtered in Wireshark, you may move on to Section D.

D. Using Wireshark to Analyze Packets (Lab Report)

1. Find and select a **GET** HTTP packet. Then, click the **+** next to **Internet Protocol** to display more info about the IP header of the packet. Fill out the table below and compare with your teammate.

	PCa	PCb
Source Address		
Destination Address		

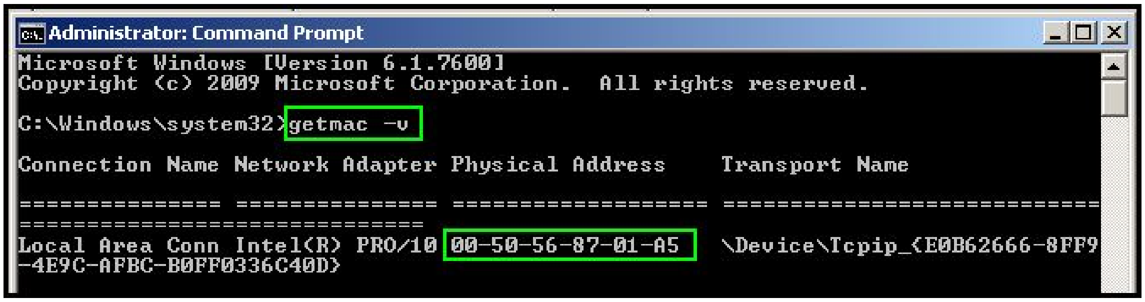
2. Now, click the **+** next to **Transmission Control Protocol** (TCP). Fill out the table below and compare once more with your teammate.

	PCa	PCb
Source Port		
Destination Port		

3. Lastly, expand the info for **Ethernet** and record your findings below.

	PCa	PCb
Source		
Destination		

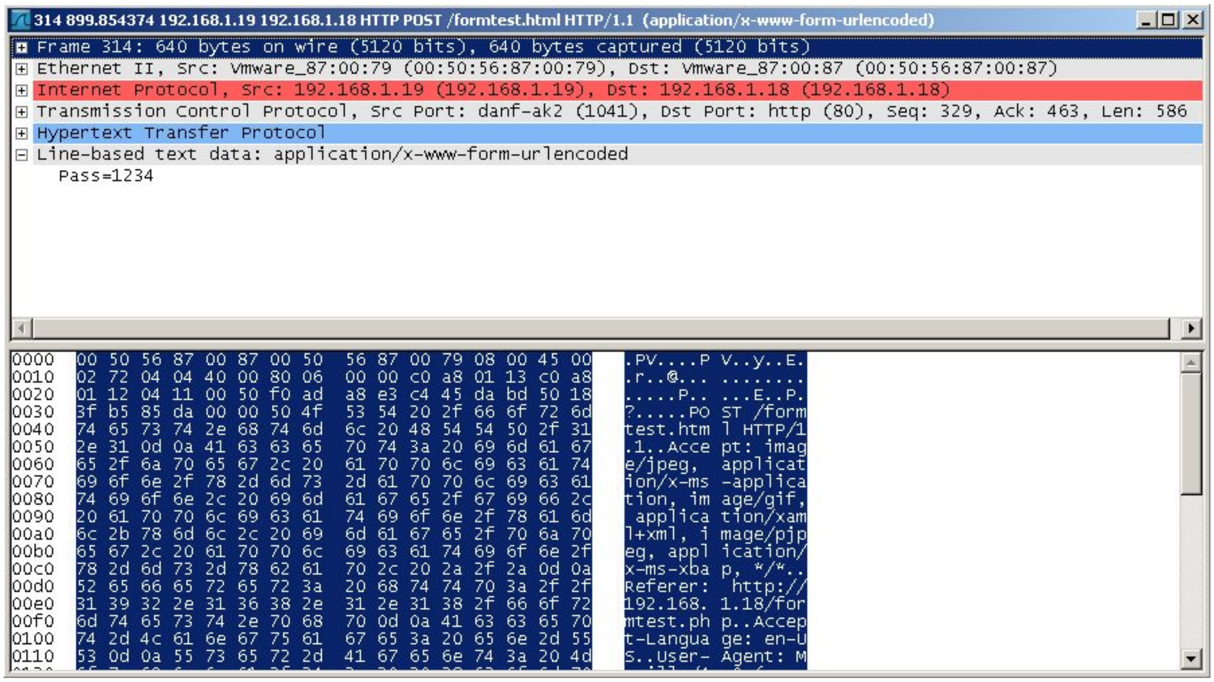
4. To follow up on what you found in the Ethernet portion of the packet, you will check your computer's **MAC (Ethernet) address** to see if it matches the Source address in the packet you analyzed. To do so, open the **Command Prompt**.
5. In the command line, type `getmac -v` and press **Enter**. This command will display the computer's MAC address for each network adapter installed. Check the **Physical Address** of the **Local Area Connection** adapter - does this match what you found in Wireshark?



Administrator: Command Prompt

```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>getmac -v

Connection Name Network Adapter Physical Address    Transport Name
=============== =============== =================== ==========================
==============================
Local Area Conn Intel(R) PRO/10 00-50-56-87-01-A5   \Device\Tcpip_{E0B62666-8FF9
-4E9C-AFBC-B0FF0336C40D}
```

6. Remember the **password** that you entered into the **form**? Back in Wireshark, see if there is any recording of password in the HTTP packets (It should be a HTTP POST packet). You should be able to locate the password in plaintext as **Pass=12345**. What does this tell you about the importance of web security?
7. Double click on a HTTP GET and HTTP POST packet. The packet will be opened as follows:

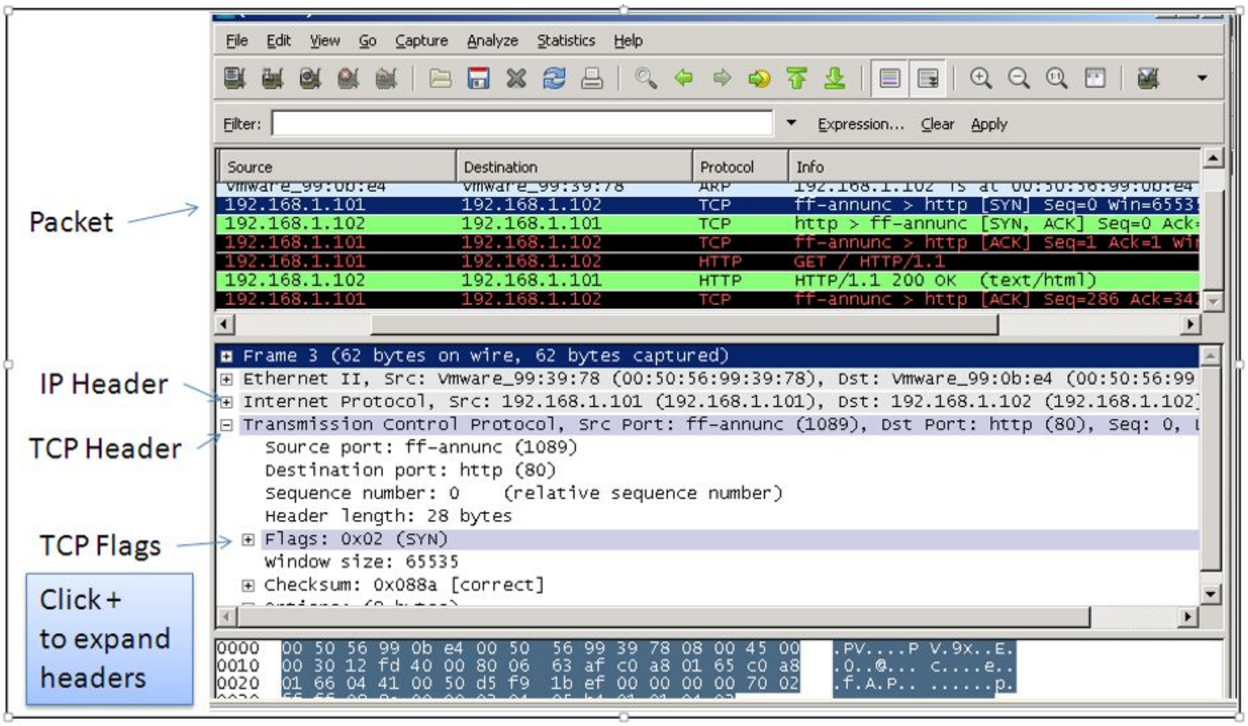


314 899.854374 192.168.1.19 192.168.1.18 HTTP POST /formtest.html HTTP/1.1 (application/x-www-form-urlencoded)

```
⊞ Frame 314: 640 bytes on wire (5120 bits), 640 bytes captured (5120 bits)
⊞ Ethernet II, Src: Vmware_87:00:79 (00:50:56:87:00:79), Dst: Vmware_87:00:87 (00:50:56:87:00:87)
⊞ Internet Protocol, Src: 192.168.1.19 (192.168.1.19), Dst: 192.168.1.18 (192.168.1.18)
⊞ Transmission Control Protocol, Src Port: danf-ak2 (1041), Dst Port: http (80), Seq: 329, Ack: 463, Len: 586
⊞ Hypertext Transfer Protocol
⊟ Line-based text data: application/x-www-form-urlencoded
    Pass=1234
```

```
0000  00 50 56 87 00 87 00 50  56 87 00 79 08 00 45 00   .PV....P V..y..E.
0010  02 72 04 04 40 00 80 06  00 00 c0 a8 01 13 c0 a8   .r..@... ........
0020  01 12 04 11 00 50 f0 ad  a8 e3 c4 45 da bd 50 18   .....P.. ...E..P.
0030  3f b5 85 da 00 00 50 4f  53 54 20 2f 66 6f 72 6d   ?.....PO ST /form
0040  74 65 73 74 2e 68 74 6d  6c 20 48 54 54 50 2f 31   test.htm l HTTP/1
0050  2e 31 0d 0a 41 63 63 65  70 74 3a 20 69 6d 61 67   .1..Acce pt: imag
0060  65 2f 6a 70 65 67 2c 20  61 70 70 6c 69 63 61 74   e/jpeg,  applicat
0070  69 6f 6e 2f 78 2d 6d 73  2d 61 70 70 6c 69 63 61   ion/x-ms -applica
0080  74 69 6f 6e 2c 20 69 6d  61 67 65 2f 67 69 66 2c   tion, im age/gif,
0090  20 61 70 70 6c 69 63 61  74 69 6f 6e 2f 78 61 6d    applica tion/xam
00a0  6c 2b 78 6d 6c 2c 20 69  6d 61 67 65 2f 70 6a 70   l+xml, i mage/pjp
00b0  65 67 2c 20 61 70 70 6c  69 63 61 74 69 6f 6e 2f   eg, appl ication/
00c0  78 2d 6d 73 2d 78 62 61  70 2c 20 2a 2f 2a 0d 0a   x-ms-xba p, */*..
00d0  52 65 66 65 72 65 72 3a  20 68 74 74 70 3a 2f 2f   Referer:  http://
00e0  31 39 32 2e 31 36 38 2e  31 2e 31 38 2f 66 6f 72   192.168. 1.18/for
00f0  6d 74 65 73 74 2e 70 68  70 0d 0a 41 63 63 65 70   mtest.ph p..Accep
0100  74 2d 4c 61 6e 67 75 61  67 65 3a 20 65 6e 2d 55   t-Langua ge: en-U
0110  53 0d 0a 55 73 65 72 2d  41 67 65 6e 74 3a 20 4d   S..User- Agent: M
```

8. **Work in this exercise with your teammate.** Expand the Internet Protocol Header and Transmission Protocol Header. Identify four fields that you think most import and their values and list them in the following table. **Discuss their functions with your teammate.**


Packet
IP Header
TCP Header
TCP Flags
Click + to expand headers
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression... Clear Apply

```
Source               Destination          Protocol  Info
Vmware_99:0b:e4      Vmware_99:39:78      ARP       192.168.1.102 is at 00:50:56:99:0b:e4
192.168.1.101        192.168.1.102        TCP       ff-annunc > http [SYN] Seq=0 Win=6553
192.168.1.102        192.168.1.101        TCP       http > ff-annunc [SYN, ACK] Seq=0 Ack=
192.168.1.101        192.168.1.102        TCP       ff-annunc > http [ACK] Seq=1 Ack=1 Wi
192.168.1.101        192.168.1.102        HTTP      GET / HTTP/1.1
192.168.1.102        192.168.1.101        HTTP      HTTP/1.1 200 OK  (text/html)
192.168.1.101        192.168.1.102        TCP       ff-annunc > http [ACK] Seq=286 Ack=34
```

```
⊞ Frame 3 (62 bytes on wire, 62 bytes captured)
⊞ Ethernet II, Src: Vmware_99:39:78 (00:50:56:99:39:78), Dst: Vmware_99:0b:e4 (00:50:56:99
⊞ Internet Protocol, Src: 192.168.1.101 (192.168.1.101), Dst: 192.168.1.102 (192.168.1.102)
⊟ Transmission Control Protocol, Src Port: ff-annunc (1089), Dst Port: http (80), Seq: 0, L
    Source port: ff-annunc (1089)
    Destination port: http (80)
    Sequence number: 0    (relative sequence number)
    Header length: 28 bytes
  ⊞ Flags: 0x02 (SYN)
    Window size: 65535
  ⊞ Checksum: 0x088a [correct]
  ⊟ Options: (8 bytes)
```

```
0000  00 50 56 99 0b e4 00 50  56 99 39 78 08 00 45 00   .PV....P V.9x..E.
0010  00 30 12 fd 40 00 80 06  63 af c0 a8 01 65 c0 a8   .0..@... c....e..
0020  01 66 04 41 00 50 d5 f9  1b ef 00 00 00 00 70 02   .f.A.P.. ......p.
```

Internet Protocol (IP) Header

Field	Value	Function

Transmission Control Protocol Header

Field	Value	Function