Romain Fontugne

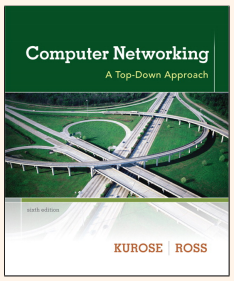
2018 Fall Semester

Information Network Systems

Wireshark Lab (TCP)

Today: Wireshark Lab 2

1. Network setup
2. Wireshark
   - HTTP
3. HTTP Client/server setup



- Wireshark Lab, from Chap. 1, 2, and 3

Network setup

Get your laptop connected to the network

- Connect a LAN cable to your laptop and the switch

Choose an IP address

- One from: 192.168.0.xxx
- xxx $\in [2, 254]$
- Should be unique!

Setup your IP address

GUI

- Type ncpa.cpl into the Start menu search box
- Click on "Local Area Connection"
- then on "Properties"
- then on "Internet Protocol Version 4 (TCP/IPv4)"
- Set your IP (192.168.0.xxx), subnet mask (255.255.255.0) and the gateway (192.168.0.1)
- Check your connectivity with your browser: http://192.168.0.1/

Setup your IP address (alternative)

Command line:

- Configure your interface with the command: netsh command
  - open a terminal with administrator privileges
  - check your current address with: netsh interface ip show config
  - netsh interface ipv4 set address name="Local Area Connection" source=static address=192.168.0.xxx mask=255.255.255.0 gateway=192.168.0.1
- Check your connectivity with the command: ping 192.168.0.1

Now you are connected!

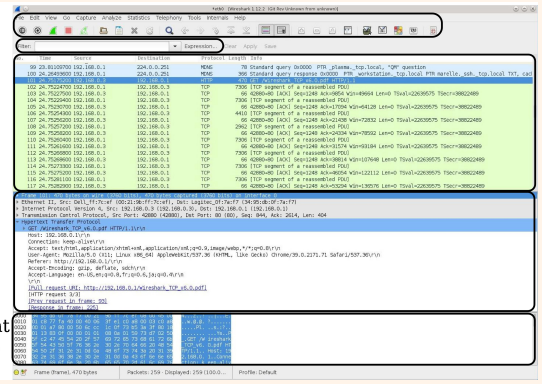
From now on:

- All documents and code are available on http://192.168.0.1/
- Open your favorite browser and type http://192.168.0.1/

Running Wireshark

Graphical User Interface:

- Command menus
- Display filter
- Listing of captured packets
- Details of selected packet header
- Packet content in hexadecimal and ASCII



TCP/HTTP file transfer

Goal: See what happen when you upload a large file to a web server

We'll need to use Wireshark to obtain a packet trace of the TCP transfer of a file from your computer to a remote server. You'll do so by accessing a Web page that will allow you to enter the name of a file stored on your computer (which contains the ASCII text of Alice in Wonderland), and then transfer the file to a Web server using the HTTP POST method. We're using the POST method rather than the GET method as we'd like to transfer a large amount of data from your computer to another computer.

TCP/HTTP file transfer

Do the following:

- Start up your web browser. From http://192.168.0.1/lab/alice.txt retrieve an ASCII copy of Alice in Wonderland. Store this file somewhere on your computer.
- Now start the capture with Wireshark.
- Go to http://192.168.0.1/lab/upload.html.
- Use the Browse button in this form to enter the name of the file on your computer containing Alice in Wonderland.
- Press the "Upload alice.txt file" button to upload the file to the server.
- Once the file has been uploaded, a short congratulations message will be displayed in your browser.
- Stop Wireshark packet capture.

First look at the trace

Before analyzing the behavior of the TCP connection in detail, let's take a high level view of the trace.

- Filter the packets displayed in the Wireshark window by entering "tcp" into the display filter
- What you should see is series of TCP and HTTP messages between your computer and the server. You should see the initial three-way handshake containing a SYN message, and an HTTP POST message. You might see a series of "HTTP Continuation" messages, this is made by Wireshark to indicate that there are multiple TCP segments being used to carry a single HTTP message.
- Since this lab is about TCP rather than HTTP, let's change Wireshark's "listing of captured packets" window so that it shows information about the TCP segments containing the HTTP messages, rather than about the HTTP messages. To have Wireshark do this, select Analyze→Enabled Protocols. Then uncheck the HTTP box and select OK.

TCP questions (1)

Answer the following questions:

1. What is the TCP port used by the client computer? and the one used by the server?
2. What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client and server?
3. What is the sequence number of the SYNACK segment sent by the server to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? How did the server determine that value?
4. What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you'll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a "POST" within its DATA field.

TCP questions (2)

Answer the following questions:

5. Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection. What is the length of each of the first six TCP segments?[a]

[a] The TCP segments in the trace file should be less that 1460 bytes. This is because Ethernet limits the length of the maximum IP packet to 1500 bytes (40 bytes of TCP/IP header data and 1460 bytes of TCP payload). If your trace indicates a TCP length greater than 1500 bytes, then Wireshark is reporting the wrong TCP segment length; it will likely also show only one large TCP segment rather than multiple smaller segments. Your computer is indeed probably sending multiple smaller segments, as indicated by the ACKs it receives. This inconsistency in reported segment lengths is due to the interaction between the Ethernet driver and the Wireshark software. If you have this inconsistency, you better have to use a hub and analyze packets sent by a different host.

TCP questions (3)

Answer the following questions:

6. What are the sequence numbers of the first six segments in the TCP connection (including the segment containing the HTTP POST)?
7. At what time was each segment sent? When was the ACK for each segment received?
8. Given the difference between when each TCP segment was sent, and when its acknowledgment was received, what is the RTT value for each of the six segments?
9. What is the minimum amount of available buffer space advertised at the received for the entire trace? Does the lack of receiver buffer space ever throttle the sender?

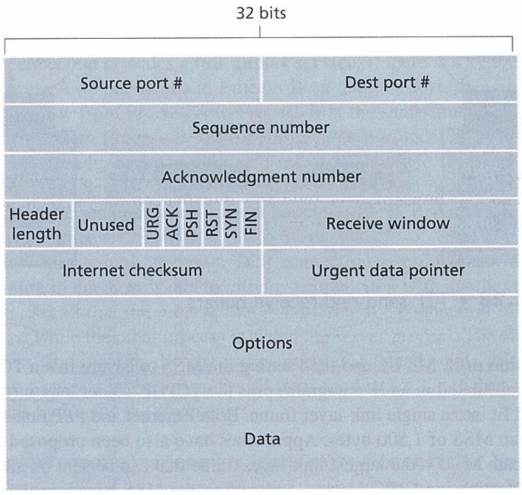
TCP questions (4)

Answer the following questions:

10. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?
11. How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing multiple received segments at once?
12. What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.

Extra

TCP: Header



Web server/client

Machine 1: web server

- Download the server code from: http://192.168.0.1/WebServer/WebServer.py
- Add files in directories where you've downloaded the server code (e.g. simple.html)
- Open a terminal, go to the directory where you've downloaded the file
- Execute the following command: python WebServer.py

Let's take a quick look at what is going on in WebServer.py

...

Machine 2: web client

- Check if the webserver works:
  - Open your favorite web browser
  - Access a file from machine1 (e.g. http://192.168.0.xxx:8080/simple.html)

HTML Documents with Embedded Objects

Now let see what happens when your browser downloads a file with embedded objects, i.e., a file that includes other objects (in the example below, image files) that are stored on another server(s).

- Start up your web browser, and make sure your browser's cache is cleared.
- Start up the Wireshark packet sniffer
- Enter the following URL into your browser http:// *IP address of machine1* /simple.html
- Your browser should display a short HTML file with images. These images are referenced in the base HTML file. That is, the images themselves are not contained in the HTML; instead the URLs for the images are contained in the downloaded HTML file.
- Stop Wireshark packet capture, and enter "http" in the display-filter-specification window.

HTML Documents with Embedded Objects

Answer the following questions:

- How many HTTP GET request messages did your browser send? To which Internet addresses were these GET requests sent?
- Can you tell whether your browser downloaded the two images serially, or whether they were downloaded from the two web sites in parallel? Explain.