Romain Fontugne

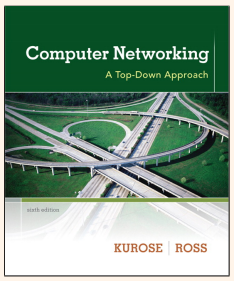
2018 Fall Semester

Information Network Systems

Wireshark Lab (HTTP)

Today: Wireshark Lab 1

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/

Wireshark: Getting Started

Wireshark is a **packet sniffer:**

- Basic tool to observe messages exchanged on a network
- Composed of two things:
  - packet capture: copy every link-layer frame received/sent by your computer
  - packet analyzer: display the contents of all protocols fields and data
- Passive monitoring: only read packets, packet sniffer never send packets

Getting Wireshark

Install Wireshark (if it is not already done)

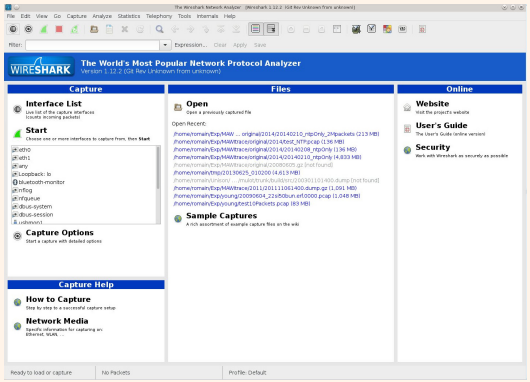
- download the binary file from http://192.168.0.1/download/wireshark
- install it on your laptop

Let's now have a quick look at wireshark

(you can also read Wireshark_Intro_v6.0.pdf from http://192.168.0.1/)

Running Wireshark

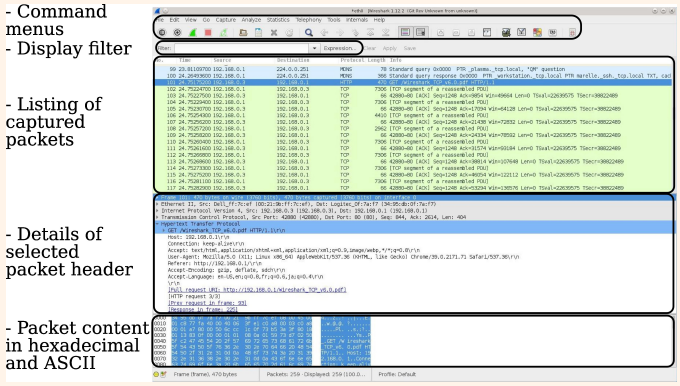
Initial Screen:



→ 2 important things: Interface list and start button

Test Run

Start then stop the capture:



Wireshark: HTTP

The Basic HTTP GET/response interaction

Let's begin our exploration of HTTP by downloading a very simple HTML file that contains no embedded objects. Do the following:

- Start up your web browser.
- Start up the Wireshark packet sniffer. **Enter "http"** (just the letters, not the quotation marks) in the display-filter-specification window and **click on "Apply"**, so that only captured HTTP messages will be displayed later in the packet-listing window.
- Start Wireshark packet capture
- Enter the following to your browser http://192.168.0.1/lab/oneline.html
- Your browser should display the very simple HTML page.
- Stop Wireshark packet capture.

Wireshark: HTTP

You captured at least 2 HTTP messages (GET and reply). Answer the following questions:

- Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running?
- What languages (if any) does your browser indicate that it can accept to the server?
- What is the status code returned from the server to your browser?
- When was the HTML file that you are retrieving last modified at the server?
- How many bytes of content are being returned to your browser?
- By inspecting the raw data in the packet content window, do you see any headers within the data that are not displayed in the packet-listing window? If so, name one.

HTTP CONDITIONAL GET/response interaction

Recall that most web browser perform caching and conditional GET. Let see how it works. Do the following:

- Start up your web browser, and make sure your browser's cache is cleared.
- Start up the Wireshark packet sniffer
- Enter the following to your browser: http://192.168.0.1/lab/twolines.html
- Enter the same URL again in a new tab (or hit the refresh button)
- Stop Wireshark packet capture, and enter "http" in the display-filter-specification window.

HTTP CONDITIONAL GET/response interaction

Answer the following questions:

- Inspect the contents of the first HTTP GET request from your browser to the server. Do you see an "IF-MODIFIED-SINCE" line in the HTTP GET?
- Inspect the content of the server response. Did the server explicitly return the contents of the file? How can you tell?
- Now inspect the contents of the second HTTP GET request from your browser to the server. Do you see an "IF-MODIFIED-SINCE" line in the HTTP GET? If so, what information follows the "IF-MODIFIED-SINCE" header?
- What is the HTTP status code and phrase returned from the server in response to this second HTTP GET? Did the server explicitly return the contents of the file? Explain

Retrieving Long Documents

So far, the documents retrieved have been simple and short HTML files. Let's next see what happens when we download a long HTML file. Do the following:

- 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://192.168.0.1/lab/billofrights.html
- Your browser should display the rather lengthy US Bill of Rights
- Stop Wireshark packet capture, and enter "http" in the display-filter-specification window, so that only captured HTTP messages will be displayed

Retrieving Long Documents

Answer the following questions:

- How many HTTP GET request messages did your browser send? Which packet number in the trace contains the GET message for the Bill or Rights?
- Which packet number in the trace contains the status code and phrase associated with the response to the HTTP GET request?
- What is the status code and phrase in the response?
- How many data-containing TCP segments were needed to carry the single HTTP response and the text of the Bill of Rights?

Extras

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.

TCP

Next time we'll investigate the behavior of the celebrated TCP protocol in detail.