Wireshark

Introduction to Wireshark (version 3.x)

Wireshark is a software protocol analyser, or "packet sniffer" application, used for network troubleshooting, analysis, software and protocol development, and education. As data streams travel back and forth over the network, the sniffer "**captures**" each protocol data unit (PDU) and can decode and analyse its content according to the appropriate RFC or other specifications.

Wireshark is also one of the tools of choice for network hackers!

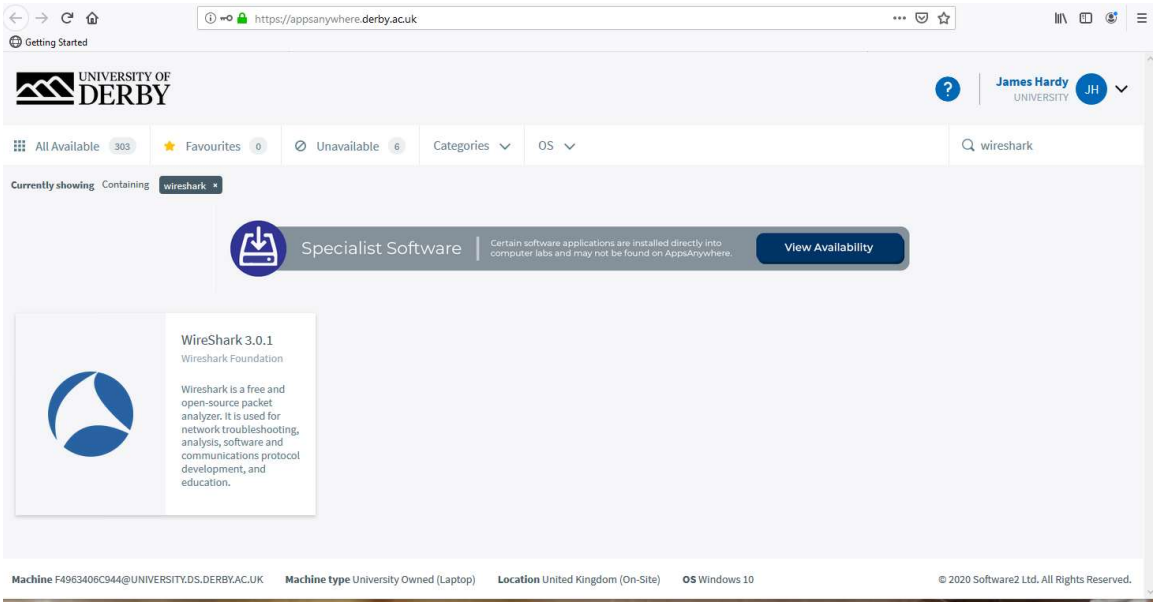
Objectives

In this tutorial you will learn how to start and use the tool to perform basic network traffic capture.

How to launch Wireshark in MS316-319 (Large PC)

1. Double-click on “Apps Anywhere” located in the lab PC’s desktop;
2. Wait until you are successfully validated;
3. Search for “wireshark”;
4. Select Launch (green “launch” button appears when mouse-over the Wireshark tile)

On the large PC you will see packets from the university network and Internet



How to launch Wireshark in MS316-319 (Small PC)

From the menu find the wireshark application

On the small PC you will only see network traffic inside the room that you are in. This is bad and good at the same time. Bad because you only see the classroom data, not the University and internet traffic. Good because you only see the classroom traffic and not the University babble!

How to launch Wireshark at home

Wireshark is free and available from the download page of the wireshark website:
https://www.wireshark.org/download.html

Simply choose the version that matches your system, download and follow the instruction to install.

If you use Linux, wireshark is generally available in the repos. Be careful, I have noticed that after initially installing wireshark there may be a delay before wireless reconnects. You might need to run wireshark with elevated privilege (sudo) from the command line (even though it asks if users should be able to use it and it installs in a menu) unless you add the current user to the "wireshark" group and reboot [sudo usermod -aG wireshark $(whoami)]

As a general comment, avoid the 32bit versions unless you have a good reason to use them.

Basics of Wireshark

The opening screen of Wireshark version 2 and 3 are very similar, showing two main sections, as illustrated in Figure 1(1) and Figure 1(2). If you have not saved any captures before, you won't see Figure 1(1)

.

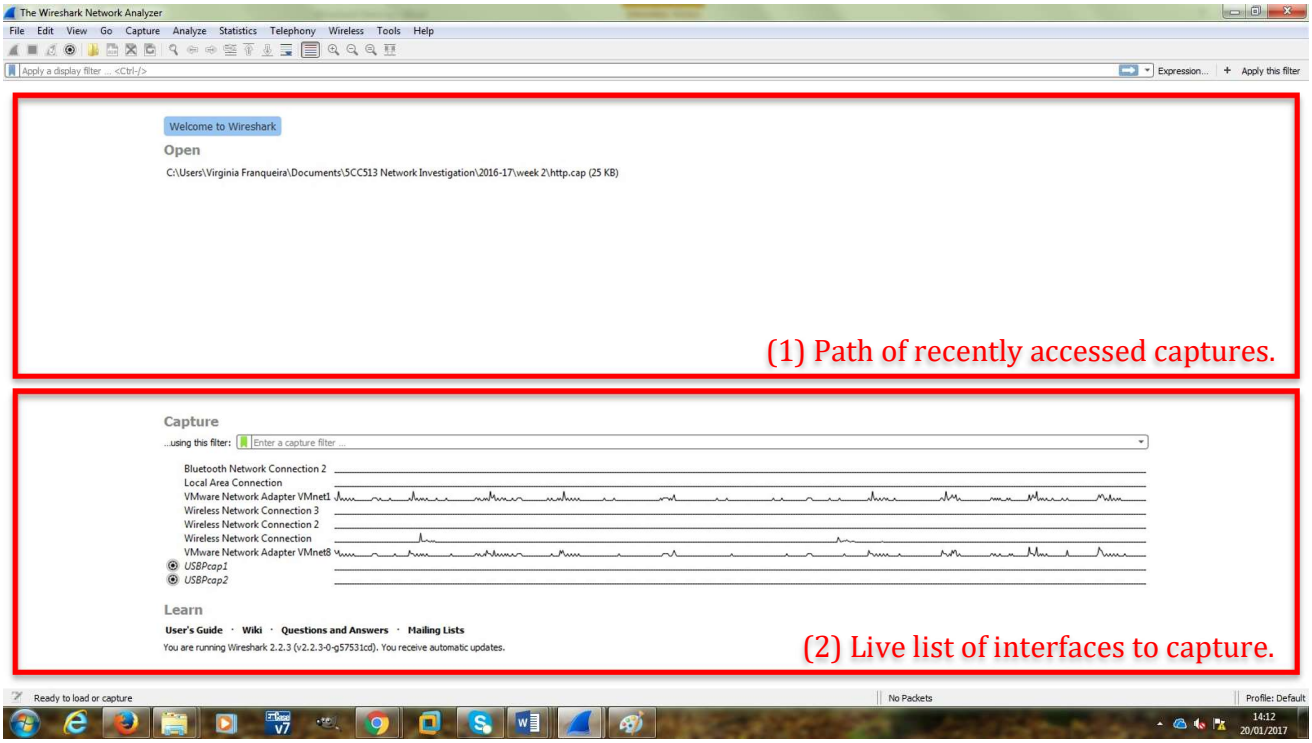


Figure 1: Opening page of Wireshark 2 / 3

Each line shown on the Figure 1(2) is an interface that is accessible to Wireshark (available for capture). The line also gives an indication of the amount of traffic that has been seen on each interface.

Capture Options

The **Capture Options** button (highlighted in Figure 2), also accessible under the menu "Capture", opens the "Capture Interfaces" screen (Figure 3) and allows you to configure advanced options.
Advanced options are distributed in 3 tabs (Input, Output and Options) and allow, e.g.:

- To determine the file where captured data will be written to.
- To resolve MAC addresses and DNS names.
- To limit the time or size of the capture.

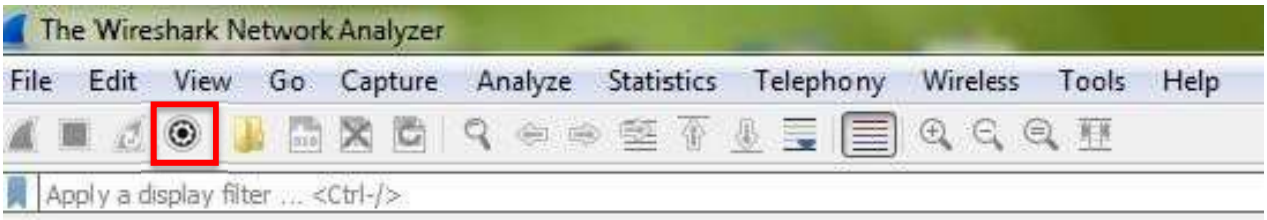


Figure 2: Capture Options button highlighted

The selection of these options helps to improve the performance of Wireshark capture. For example, resolving names requires a network lookup which can be very slow; there will be a substantial improvement in performance when this is disabled but at the expense of not seeing human readable names (if they exist). Name lookups generate large numbers of name queries.

Time and size limits can also place limitations especially on unattended captures, this can be changed on the Output tab and limited on the Options tab.

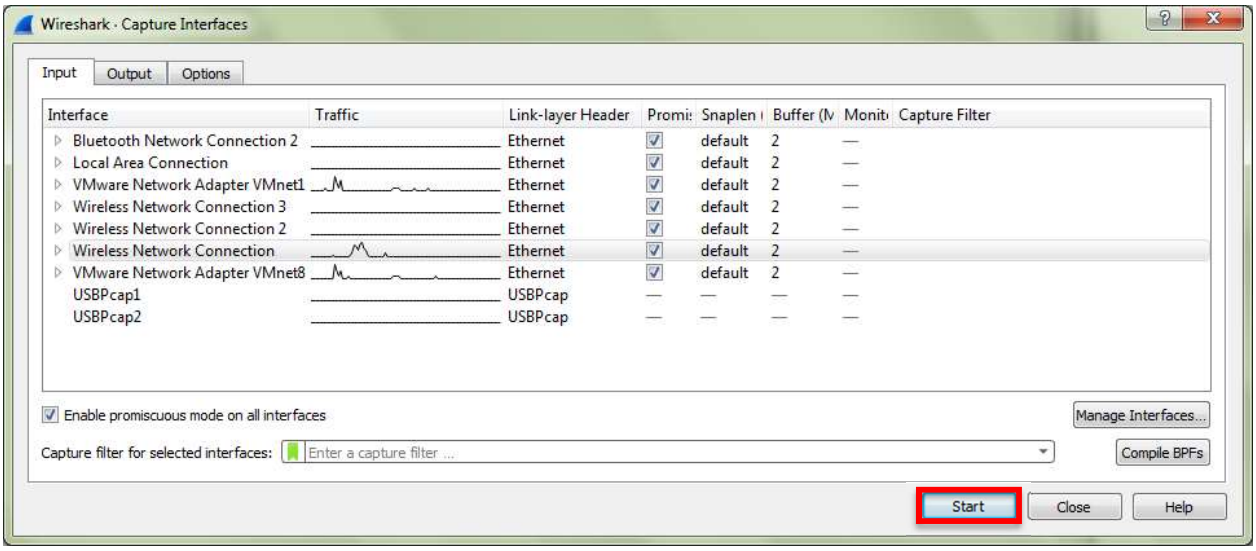


Figure 3: The Capture Interfaces screen.

Feel free to explore the tabs, after the next section you can return and test the results of some of the options.

Running a simple packet capture

You can select an interface to observe directly from the Options menu or the main screen without using the Options screen. The process is similar in either case.

Select one interface which is showing variation in traffic flow, i.e., choose one of wiggly lines from Figure 1(2) or Figure 3 rather than the flat lines. Look at the description to confirm your choice; your PC is probably connected to the network using either a "Local Area Connection", "Wifi" or "wireless network connection", these would be good choices for this tutorial. The name is taken from the network

interfaces defined on the local machine. The "VMware" interfaces typically provide a link between the real and virtual worlds (you will have many opportunities to experiment with these at a later stage).

You can start the capture by double-click on the interface or select the interface with a single click and then choose "start" from the options screen. From the main screen you can use "Start" from the "capture" menu shown in **Error! Reference source not found.**, the start (blue sharkfin) icon on the menu bar or double click an interface.

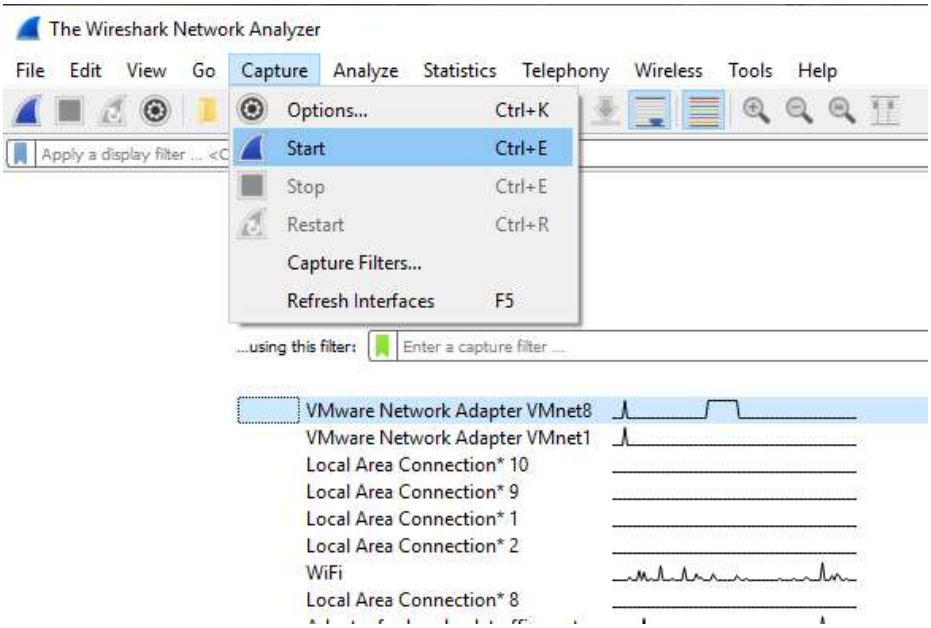
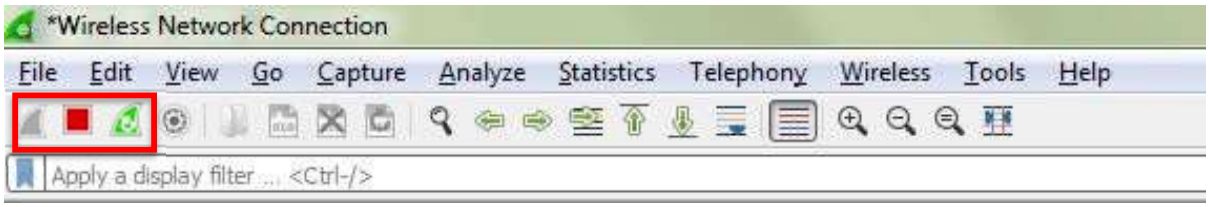


Figure 4: Capture / start option

The stop icon is the square next to the sharkfin, it will become red when a capture is in progress. Stop capturing live packets is possible via the red button (2nd highlighted in Figure 5). To restart the current capture, use the green button (3rd highlighted in Figure 5). All these
options are also available via the **Capture** menu.

Figure 5: Manipulating captures – Start, Stop and Restart, respectively.



If you stop a capture, and either want to start a new one or restart the same capture, Wireshark will ask if you want to save the captured packages or not.

Start a capture using one of the methods stated.

Wireshark screen will immediately begin filling up with traffic seen on the network interface and should look something like **Error! Reference source not found.**.

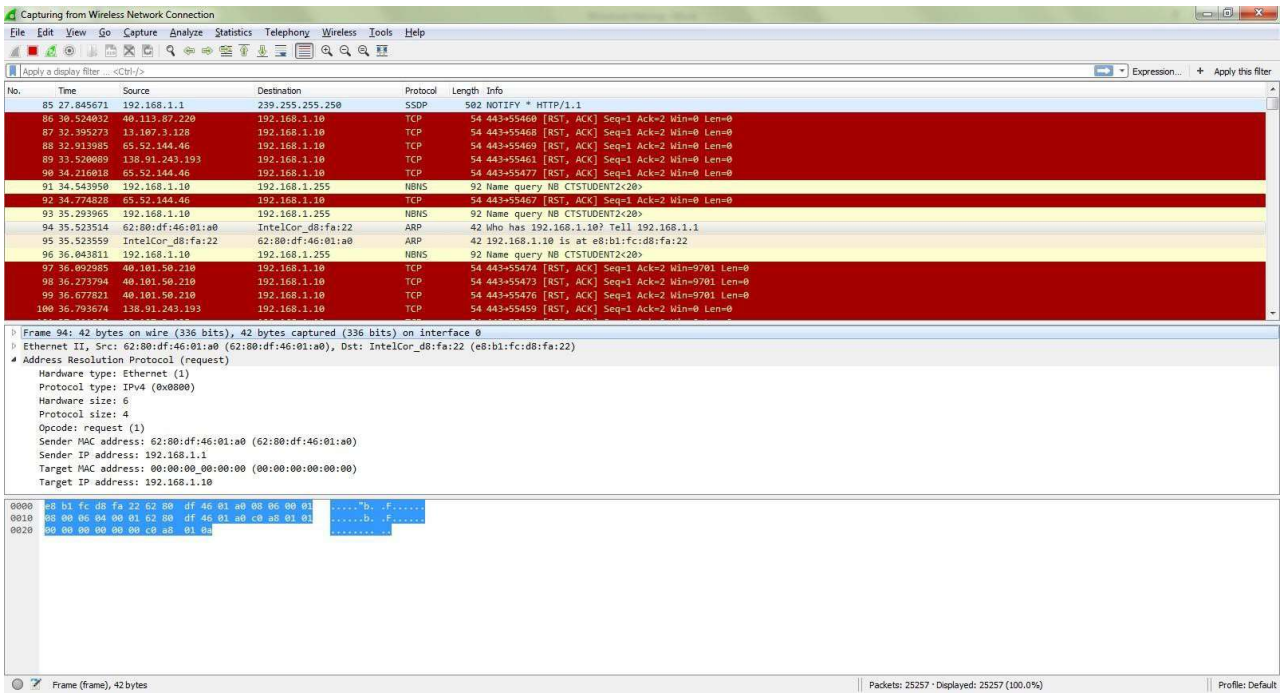


Figure 6: Live capture of packets on the selected interface

Leave the capture running for a few minutes or until the top pane is full. If there is no data, stop the capture and check which interface you are using.

Stop the capture

Each line in the top pane of the Wireshark window corresponds to a **single packet** seen on the network. The default display shows:

- The time of the packet (by default this is relative to the initiation of the capture, but

can be changed – NTP becomes very important!)
- The source and destination IP addresses,
- The protocol used and some information about the packet.

You can see more information by clicking on a row. This causes the bottom two window panes to fill with information:
- The middle pane contains the details of the packet selected in the top frame.
- Clicking The ">" icons reveal varying levels of detail about each layer of information contained within the packet.

Select a packet, and examine its content across the top, middle and bottom panes.

The following link contains useful information about setting up capture of network traffic. It starts with a warning – *you need to make sure you are allowed to capture packets from the network in the first place.*

https://wiki.wireshark.org/CaptureSetup

Wireshark colouring scheme

Colour coding is helpful when analysing packets with Wireshark. Notice in the capture you have done that each row is colour-coded, according to different protocols.

Wireshark uses a complex colouring scheme (which you can customize). The default settings can be accessed via the **View menu → Coloring Rules**. The default scheme is shown in Figure 7.

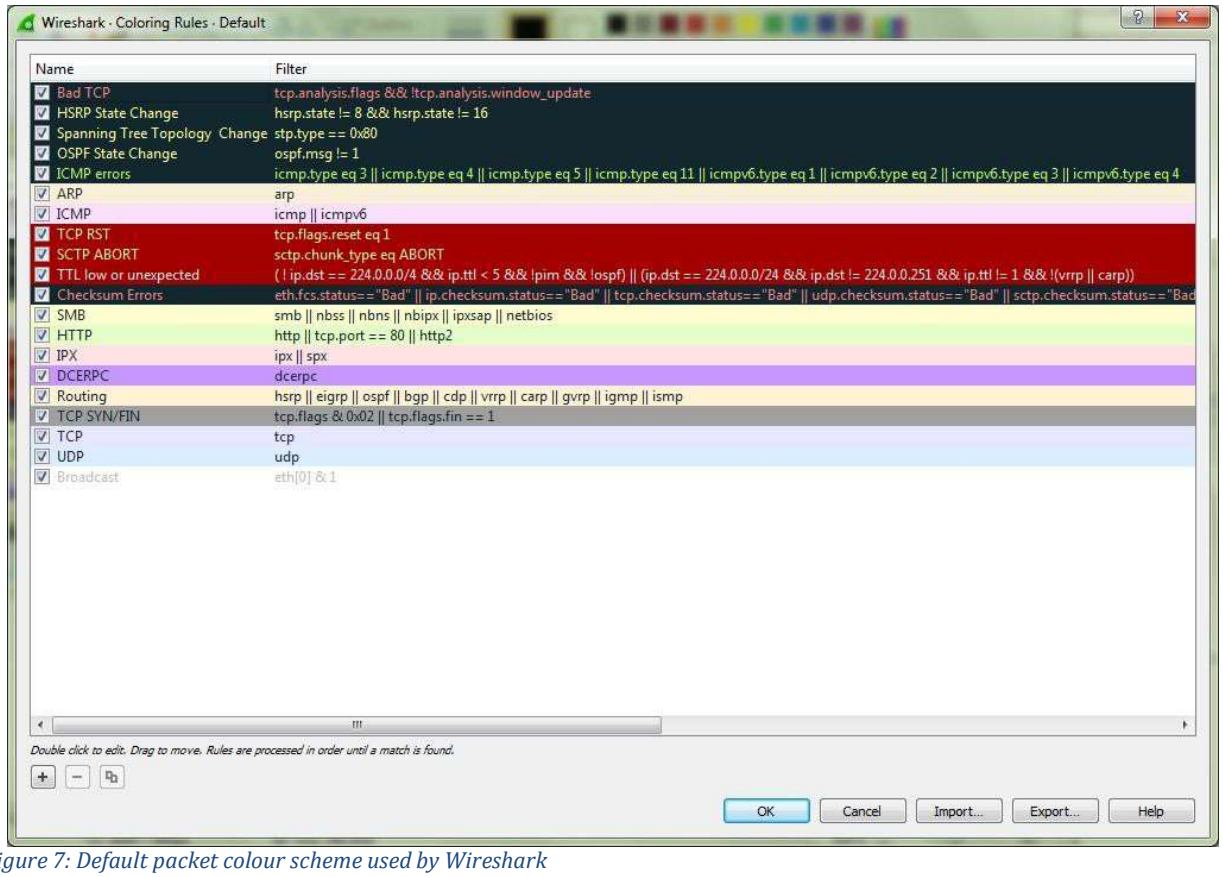


Figure 7: Default packet colour scheme used by Wireshark

Packet display filtering

You can apply this section to the data that you have captured or a sample file provided by Wireshark. Download the Wireshark capture file **http.cap** from the website: http://wiki.wireshark.org/SampleCaptures

Open the **http.cap** file via menu **File → Open**

You will see the saved packets as shown in Figure 8.

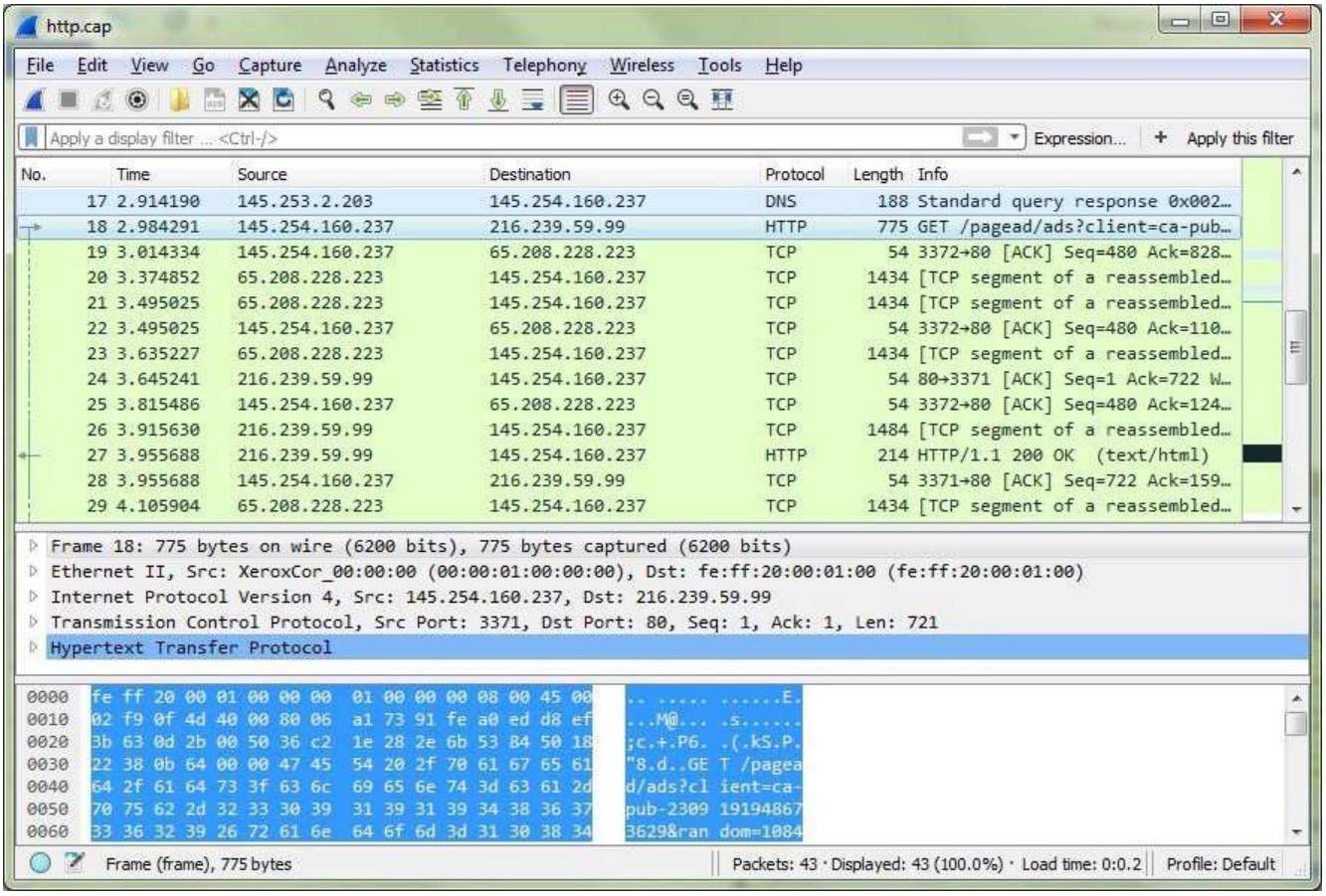


Figure 8: Capture file http.cap open on Wireshark

It is possible (crucial!) to create filters that show only packets according to a filtering rule to facilitate the analysis of a given capture. This will isolate the particular exchange or the analysis of a specific protocol. For that, we use the **Filter** section in the top bar, highlighted in Figure 9.

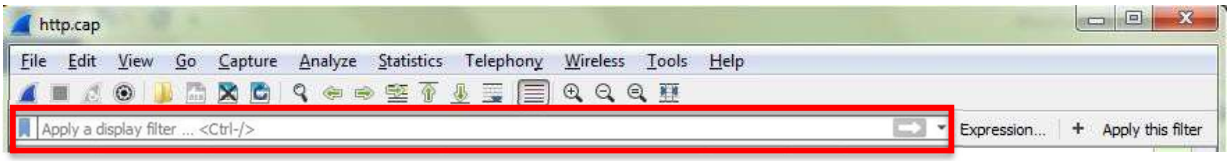


Figure 9: Feature which allow filtering packets for display purposes.

You can create filtering rules. A rule is based on the different packet header fields for known protocols.

- Type the rule **http** (under "*Apply a display filter*") and either click on the (blue) forward arrow on the right or just press **Enter**.

The number of packets displayed will reduce from 43 to 4.

Click on the "X" (red) button that appears before the arrow to remove the filtering rule.

- Type the rule **ip.dst==145.254.160.237** and press **Enter**. The number of packets displayed will reduce from 43 to 23. Remove the filtering rule.

You can create multiple combined filtering rules using the following operators:

- **&&** (AND)
- || (OR)
- **!** (NOT)

For example, if we want all packets with an IP destination equal to 145.254.160.237 and with a source or destination port different from 80:

- Type the rule **ip.dst == 145.254.160.237 && !tcp.port == 80** and press **Enter**. The number of packets displayed will reduce from 43 to 1. Remove the filtering rule.

Read more about *Display Filtering* and examples in:

https://wiki.wireshark.org/DisplayFilters

You can also use the **Expression** button, highlighted in Figure 10 to build display filters based on packet header fields of 2000 protocols recognised by Wireshark. For more details, refer to:

https://www.wireshark.org/docs/dfref/

Figure 10: Expression button helpful to build complex filters for all protocols recognised by Wireshark

Close the http.cap file via **File → Close**

End of Tutorial