

FIT1047 Tutorial 9 – Sample Solution

Topics and goals

- Network layers and protocols, learn about addresses at various layers
- Application layer – HTTP
- Familiarise yourself with some of the networking tools

Instructions

The tasks are supposed to be done in groups of two or three students.

Task 1: Basic knowledge

Briefly explain the following terms:

1. Bit rate, bandwidth

Bit rate: the numbers of bits that can be transmitted over a data communications channel per time unit, usually measured in bits per second (or kilobits, megabits, gigabits per second). Bandwidth: strictly speaking, the width of a range of frequencies (i.e. highest minus lowest frequency). E.g. each channel in WiFi has a bandwidth of 20 MHz. But it's often used as a synonym for bit rate.

2. Latency

The time it takes for a message from being sent to being received. Note that a network can have a very high bit rate but still feel slow: e.g. consider using a satellite connection for a video conference: you can stream very high quality video over the satellite connection, but due to the high latency, you could experience a delay of several seconds, which makes the connection almost unusable for real-time communication. This is also often a problem when playing real-time online video games.

3. Networks layers, PDU, encapsulation

Task 2: Physical layer

How long is a bit? This sounds like a silly question, but we can actually compute it.

The speed of light is approximately 3×10^8 m/s. But signals in copper cables are slower, they travel with approximately 70% of the speed of light.

Now if we assume that bits are transmitted sequentially through a copper cable (one after the other), and if we know how many bits are transmitted per second, we can calculate how long the signal is (in meters) for each bit!

Compute the length of a bit for the original Ethernet, which ran at 10 Mbit/s.

$$\frac{0.7 \times 3 \times 10^8 \text{ m/s}}{10 \times 10^6 \text{ bits/s}} = 21 \text{ m/bit}$$

Task 3: Packet sniffing

A packet analyser (sometimes also called “packet sniffer”) is a program that can log all packets that are received and transmitted over a network interface. We will be using *Wireshark*, a very popular open-source tool for packet analysis. You can download it on your own computer from www.wireshark.org, or use the version installed on the Monash lab computers.

This week, we will analyse a sequence of packets captured on Guido’s network at home. You can download the log file that Wireshark produced from the FIT1047 Moodle site (week 7). The file is called `Wireshark_http_example.pcap`. The diagram in Figure 1 explains how Guido’s computer is connected to the Monash web server.

1. Start Wireshark and load the capture file (see Fig. 2). After opening the file, the main window should look like the one in Fig. 3.
2. Select “frame” number 6 (as in Fig. 3). This frame shows a request sent from Guido’s home computer to the web server at www.csse.monash.edu, requesting his homepage
3. Familiarise yourself with the three main sections (panes) of the Wireshark window:
 - The packet list pane displays a summary of each packet captured. When you click on a packet here, the other two panes are updated with the details for that packet.
 - The packet details pane below shows information about the selected packet.
 - The packet bytes pane displays the raw data for the selected packet. It highlights the data for the field that is selected in the packet details pane.

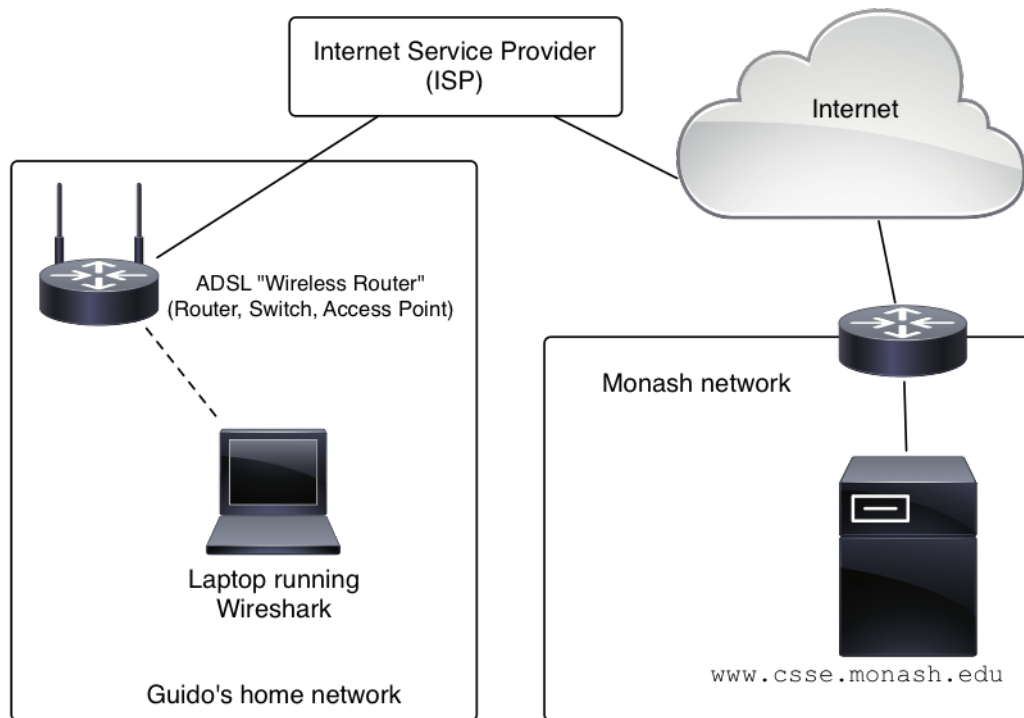


Figure 1: The network where the traffic was captured

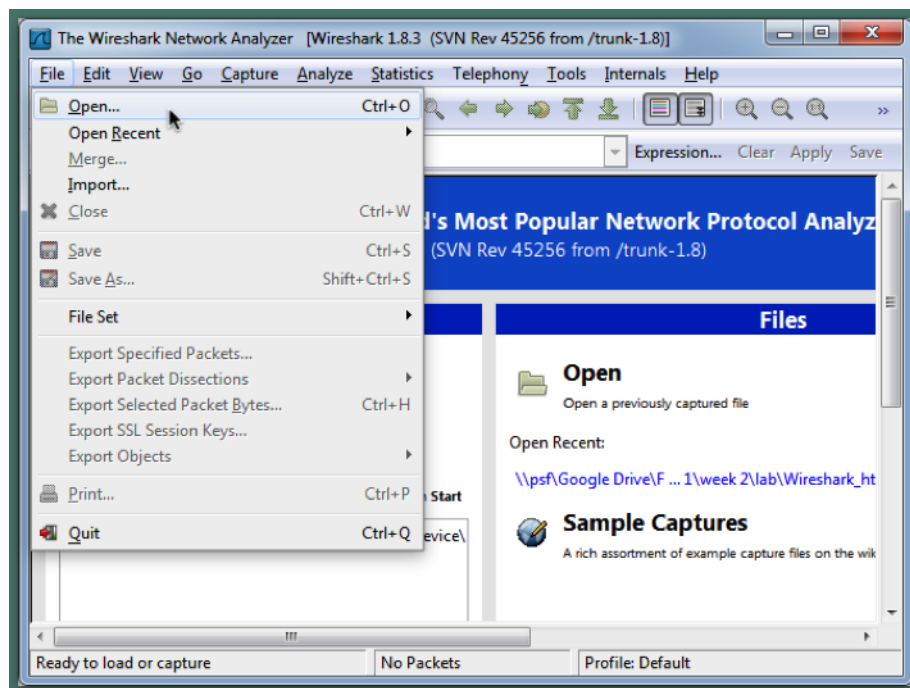


Figure 2: Wireshark File menu

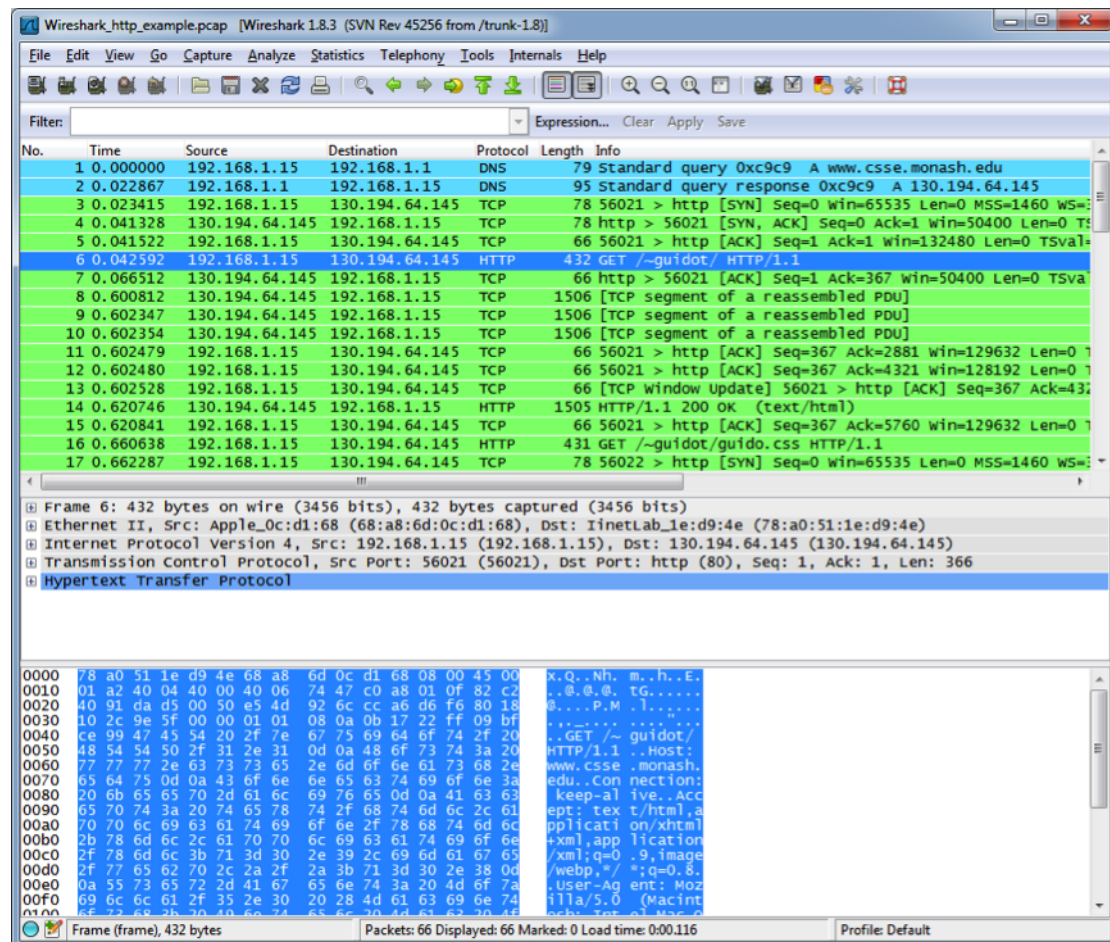


Figure 3: Wireshark main window

Navigate between the panes and explore the relationships between the displayed pieces of information.

4. Click on frame 6.

- a) How long is it in bytes?

432 bytes

- b) Which application layer protocol does it use?

HTTP

5. There are five lines in the packet details pane for frame 6, each of which can be expanded by clicking on the “+” symbol.

- a) Can you identify what they stand for?

From top to bottom: the entire frame captured by Wireshark; the data link layer; the network layer; the transport layer; the application layer.

- b) Which protocols are being used?

From top to bottom: Ethernet; IP version 4; TCP; HTTP.

- c) What are the names of the PDUs for each protocol?

Ethernet: frame; IP: packet; TCP: segment; HTTP: message.

- d) Recall that each protocol layer encapsulates the message from the layer above and adds a header. What are the sizes of the headers for each PDU used in frame 6?

You can find the size by clicking on the relevant protocol line and counting the number of bytes highlighted in the bottom pane. Some protocols also tell you the size when you open the details view. Ethernet: 14 bytes; IP: 20 bytes; 32 bytes; HTTP: we can consider the entire contents to be the actual message, there are no additional headers (such as the ones for TCP or IP).

Task 4: Data link layer / Ethernet

This activity is a continuation of the packet sniffing we did in the previous activity. Now you will use **Wireshark** to capture the live network traffic into and out of your computer.

1. Open Wireshark and start capturing packets from the appropriate network interface. In the Monash labs, that would usually be an *Ethernet* interface. You can select that interface on the main Wireshark screen, or select *Interfaces* from the *Capture* menu.

2. The Wireshark window will now start filling up with packets. Open the following URL in a web browser:

`http://s-cah-mwallace.infotech.monash.edu/~guidot/image.php`

3. Answer the following questions, based on the contents of the Ethernet frame containing the HTTP GET message.

- a) What is the value of the Ethernet address of your computer? (Hint: on Windows, use the `ipconfig /all` command line tool)
- b) What is the destination address in the Ethernet frame? Is this the Ethernet address of `www.csse.monash.edu`?

The destination MAC address belongs to the router that your computer is connected to, not to `www.csse.monash.edu`.

- c) How many bytes from the very start of the Ethernet frame does the ASCII “G” in “GET” appear in the Ethernet frame?
4. Next, answer the following questions, based on the contents of the Ethernet frame containing the first byte of the HTTP response message.

- a) What is the value of the Ethernet source address? Is this the address of your computer, or of `www.csse.monash.edu`?

The source MAC address belongs to the router that your computer is connected to, not to `www.csse.monash.edu`.

- b) What is the destination address in the Ethernet frame? Is this the Ethernet address of your computer?

The destination MAC address should be your own computer.

- c) How many bytes from the very start of the Ethernet frame does the ASCII “O” in “OK” (i.e., the HTTP response code) appear in the Ethernet frame?