Part II

Signals Circuits and Systems

Workshop 4: RLC Circuits

Objective: To understand transients and frequency responses of RLC circuits.

Outcome: After successful completion of this session, the student will be able to

1. Understand transient behaviour of RLC circuits
2. Understand how the voltage and current varies across RLC circuits and their dependency on the frequency
3. Understand resonance in RLC circuits

Equipment Required:

1. A calculator
2. Signal generator
3. Oscilloscope
4. Digital multimeter
5. Breadboard and wires

Components Required:

1. Resistors - 10 $k\Omega$ (1 No.), 2.2 $k\Omega$ (1 No.), 1 $k\Omega$ (1 No.), 100 Ω (1 No.)
2. Capacitors - 47 μF (1 No.), 100 nF (1 No.)
3. Inductors - 10 mH (1 No.)

4.1 Introduction

RLC circuits consist of resistors, capacitors and inductors. Unlike resistors, inductors and capacitors are sensitive to the variations in current and voltage. Hence, they show frequency-sensitive behaviours. Consequently, the input-output relationship of an RLC circuit will depend on the input frequency, among the other factors. In this experiment, we will observe how different configurations of resistors, capacitors and inductors respond to different input signals.

4.2 Pre-Lab

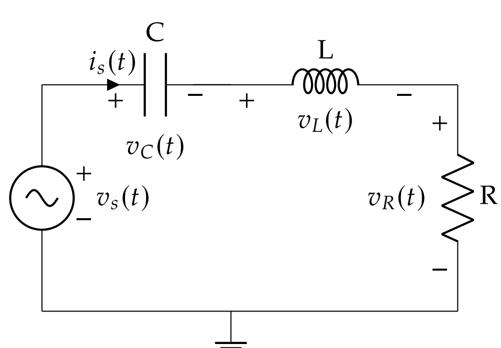

Figure 4.1: RLC filter

Task 1. *Consider the RLC circuit illustrated in Figure 4.1. Let the source frequency be f. Using the relationships, $v_C(t) = \frac{1}{j2\pi f C} i_C(t)$, $v_L = j2\pi f L\ i_L(t)$ and $v_R(t) = R\ i_R(t)$, obtain an expression for $v_R(t)$ in terms of $v_s(t), R, L, C$ and f.*

Task 2. *Calculate the magnitude of $v_R(t)$ in terms of $|v_s(t)|, R, L, C$ and f.*

Task 3. *From the result obtained in Task 2, find the frequency $\hat{f}$ for which the $|v_R(t)|$ is maximized.*

Task 4. *Calculate the phase of $v_R(t)$ in terms of $\angle v_s(t), R, L, C$ and f. Obtain the value of $\angle v_R(t)$ when $f = \hat{f}$.*

4.3 Transients

Transient responses are the changes in the output observed immediately after a sudden change in the input. They die out slowly allowing the system to settle down to an equilibrium state. Under this section, we will observe the transient response of a capacitor and an inductor for a step signal (a sudden increase/decrease of voltage).

4.3.1 Transient Response of a Capacitor

Task 5. *Construct the circuit illustrated in Figure 4.2 on the breadboard.*

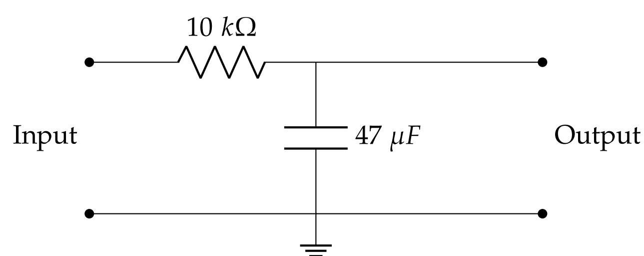

Figure 4.2: Circuit for observing the transient response of a capacitor

Task 6. *Apply a $0.1\ Hz$, $5\ V$ peak-to-peak square wave with a DC offset of $2.5\ V$ to the input using the signal generator. Observe the input and output waveforms from the oscilloscope (set the time scale of the oscilloscope to $2.5\ s$). Draw one period of the output wave.*

4.3.2 Transient Response of an Inductor

Task 7. *Construct the circuit illustrated in Figure 4.3 on the breadboard.*

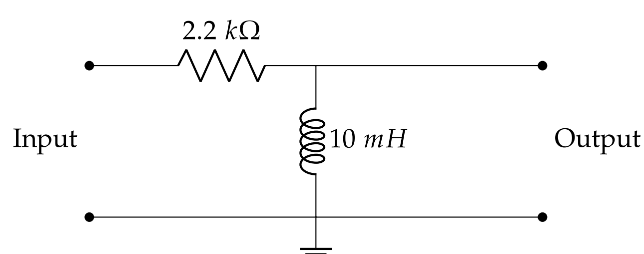

Figure 4.3: Circuit for observing the transient response of an inductor

Task 8. *Apply a $10\ kHz$, $2\ V$ peak-to-peak square wave with a DC offset of $1\ V$ to the input using the signal generator. Observe the input and output waveforms from the oscilloscope (set the time scale to $2.5\ \mu s$). Draw one period of the output wave.*

4.4 RC Filters

RC filters consist of capacitors and resistors. They better suit for driving high impedance loads. In this section we will observe their magnitude and phase responses for sinusoidal inputs.

4.4.1 RC Low-pass Filter

Task 9. *Construct the circuit illustrated in Figure 4.4.*

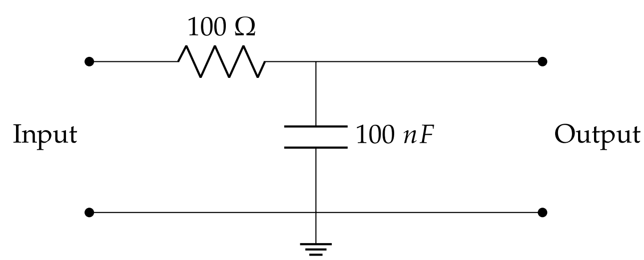

Figure 4.4: RC low-pass filter

Task 10. *Connect the two probes of the oscilloscope to the input and the output of the above circuit. Provide a* 2 V *peak-to-peak sinusoidal input signal with following frequencies and record the input and the output amplitudes along with the voltage gains in Table 4.1 given below.*

Frequency	Input amplitude	Output amplitude	Voltage gain
1 kHz			
10 kHz			
100 kHz			

Table 4.1: RC low-pass filter magnitude response

Task 11. *Observe and comment on how the phase difference between the input and the output waveforms changes with the frequency.*

Task 12. *Measure the phase difference between the input and output waveforms when the input frequency is* 100 kHz *(use **cursors** in the oscilloscope). Show your calculations.*

4.4.2 RC High-pass Filter

Task 13. *Construct the circuit illustrated in Figure 4.5.*

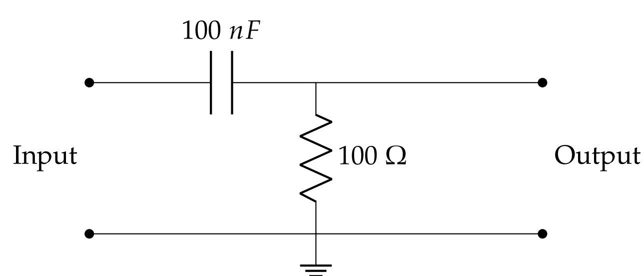

Figure 4.5: RC high-pass filter

Frequency	Input amplitude	Output amplitude	Voltage gain
1 kHz			
10 kHz			
100 kHz			

Table 4.2: RC high-pass filter magnitude response

Task 14. *Connect the two probes of the oscilloscope to the input and the output of the above circuit. Provide a* 2 V *peak-to-peak sinusoidal input signal with following frequencies and record the input and the output amplitudes along with the voltage gains in Table 4.2 given below.*

Task 15. *Use the* ***math*** *function in the oscilloscope to observe the voltage across the capacitor, when the input frequency is* 10 kHz. *Draw all the three waveforms (to the same scale) on the same plot.*

Task 16. *Observe and comment on how the phase difference between the input and output waveforms changes with the frequency.*

4.5 RL Filters

RL filters consist of inductors and resistors. They better suit for driving low impedance loads. In this section we will observe their magnitude and phase responses for sinusoidal inputs.

4.5.1 RL Low-pass Filter

Task 17. *Construct the circuit illustrated in Figure 4.6.*

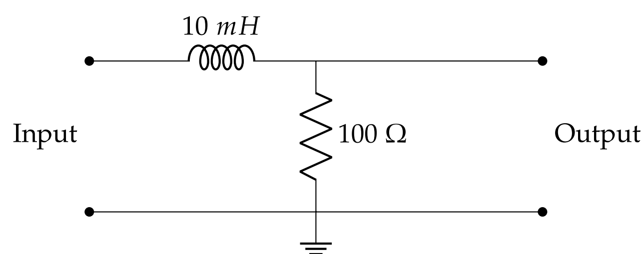

Figure 4.6: RL low-pass filter

Task 18. *Connect the two probes of the oscilloscope to the input and the output of the above circuit. Provide a* 2 V *peak-to-peak sinusoidal input signal with following frequencies and record the input and the output amplitudes along with the voltage gains in Table 4.3 given below.*

Frequency	Input amplitude	Output amplitude	Voltage gain
1 kHz			
10 kHz			
100 kHz			

Table 4.3: RL low-pass filter magnitude response

Task 19. *Observe and comment on how the phase difference between the input and output waveforms changes with the frequency.*

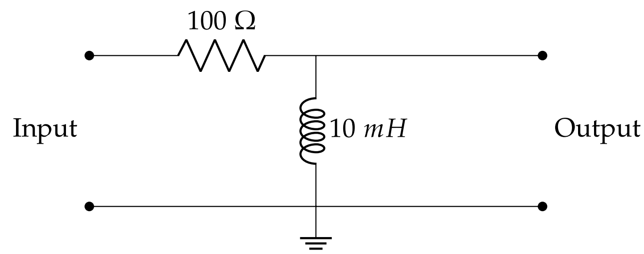

Figure 4.7: RL high-pass filter

4.5.2 RL High-pass Filter

Task 20. *Construct the circuit illustrated in Figure 4.7.*

Task 21. *Connect the two probes of the oscilloscope to the input and the output of the above circuit. Provide a 2 V peak-to-peak sinusoidal input signal with following frequencies and record the input and the output amplitudes along with the voltage gains in Table 4.4 given below.*

Frequency	Input amplitude	Output amplitude	Voltage gain
1 kHz			
10 kHz			
100 kHz			

Table 4.4: RL high-pass filter magnitude response

Task 22. *Observe and comment on how the phase difference between the input and output waveforms changes with the frequency.*

4.6 RLC Filters

RLC circuits act as cascaded RL, RC circuits and hence show combined behaviours including resonance. This section focuses on observing their magnitude and phase responses for sinusoidal inputs.

Task 23. *Construct the circuit illustrated in Figure 4.8 on the breadboard.*

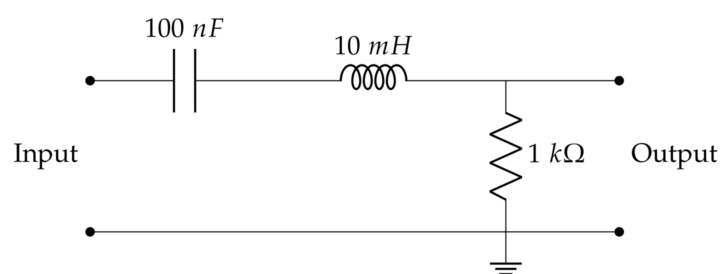

Figure 4.8: RLC filter

Task 24. *Connect the two probes of the oscilloscope to the input and the output of the above circuit. Provide a 2 V peak-to-peak sinusoidal input signal with following frequencies and record the input and the output amplitudes along with the voltage gains in Table 4.5.*

Task 25. *Suggest a method to find out the resonance frequency of the above circuit using the result of Task 4.*

Task 26. *Record the resonance frequency, $\hat{f}$. Calculate the theoretical value of $\hat{f}$ from the result obtained in Task 3.*

Frequency	Input amplitude	Output amplitude	Voltage gain
1 kHz			
10 kHz			
100 kHz			

Table 4.5: RLC filter magnitude response

Task 27. *Verify the above result by observing the magnitude variation of the output while varying the input frequency within a small range around* $\hat{f}$.

♣ The End ♣

Part III

Electronics

Workshop 4: Bipolar Junction Transistor (BJT) Amplifier

Objective: To design a simple transistor amplifier using a BJT

Outcome: After successful completion of this session, the student will be able to,

1. Test a BJT
2. Build a simple BJT amplifier with a fixed bias current
3. Use the oscilloscope to observe and measure amplifier input and output voltage waveforms
4. Estimate the voltage gain of the amplifier
5. Identify the waveform distortion resulting from over-drive
6. Waveform distortion due to improper bias point

Equipment Required:

1. DC power supply
2. Digital Oscilloscope
3. Signal generator
4. Analog multimeter
5. Breadboard and wires

Components Required:

1. BJT: 2N2222
2. Capacitors: 4.7μF (2 nos.)
3. Resistors: 560 Ω, 10 kΩ, 470 kΩ

4.1 Testing a Bipolar Junction Transistor

Task 1. *Observe the 2N2222 BJT and identify its terminals using the data sheet. Draw the pin-out diagram.*

Task 2. *Is it a pnp or an npn transistor?*

A BJT has two p-n junctions. Base is the common layer for both the junctions. As each junction behaves somewhat similar to a diode, their proper functionality can be tested by following steps similar to checking a diode.

Task 3. *Mention the colour of the multimeter probe (in diode mode) that needs to be connected to the anode of a diode, in-order to check it under forward bias configuration.*

Task 4. *Perform the multimeter test to check whether the transistor is working or not. Use the digital multimeter in diode mode.*

4.2 Building a simple BJT amplifier with fixed bias

Task 5. *On the breadboard, construct the amplifier given in figure 4.1, using the 2N2222 transistor, which has already been tested and found to be working.*

Task 6. *Make the DC measurements to check if the transistor is properly biased for amplification of an AC signal.Take R_1=470 kΩ, R_2=560 Ω, C_1= C_2= 4.7 μF, V_{CC}= +13 V DC.*

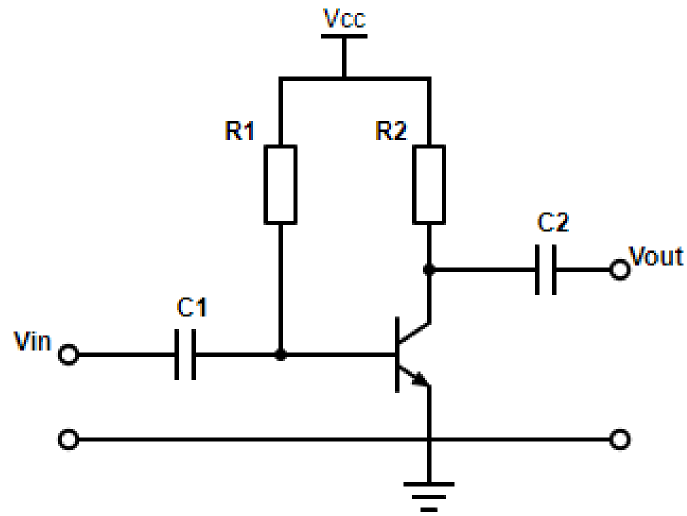

Figure 4.1: Schematic diagram of the BJT amplifier with fixed bias

We may utilize the Ohm's law to estimate the base and collector DC currents by measuring the DC voltages across each resistor using the multimeter.

Task 7. *Calculate the collector current, I_C, by measuring the voltage across R_2. Show your calculations.*

Task 8. *Calculate the base current, I_B, by measuring the voltage across R_1. Show your calculations.*

Task 9. *Measure and record the DC voltage across the collector and the emitter (V_{CE}).*

Generally, the quiescent point is selected such that $V_{CE} \approx \frac{1}{2} V_{CC}$ to allow a maximum swing of the amplified signal without causing a distortion due to saturation.

Task 10. *Considering the V_{CE} value obtained by measurement, comment on the suitability of that value for small signal amplification.*

Task 11. *What is the static or the DC current gain (h_{FE}) of this transistor? Show your calculations.*

4.3 Using the oscilloscope to observe amplifier waveforms

Task 12. *Connect a 10 kΩ resistor to the output of the amplifier circuit.*

Task 13. *Connect the signal generator to the amplifier circuit and apply the small sinusoidal AC signal as specified below. To observe the input and output waveforms, connect the channel 1 and 2 of the oscilloscope to the input and the output, respectively.*

Task 14. *In the signal generator, make adjustments to output a 30 mV (pk-pk), 100 kHz sinusoidal signal.*

Task 15. *Observe both input and output waveforms from the oscilloscope. Observe the amplifier gain and phase.*

Task 16. *What is the phase change introduced by the amplifier?*

Task 17. *Draw the input and the output voltage waveforms.*

4.4 Estimating the voltage gain of the Amplifier

Task 18. *What is the peak to peak input voltage, V_{in}?*

Task 19. *What is the peak to peak output voltage, V_{out}?*

Task 20. *What is the Voltage gain, A_V?*

4.5 Observing the waveform distortion resulting from over-drive

Task 21. *Increase the amplitude of the input signal generator and observe the distortion of the output voltage waveform due to increased drive signal or over-drive.*

Task 22. *Draw the input and output signals.*

4.6 Observing the waveform distortion due to improper bias point

Task 23. *Replace R_1 with a 1 MΩ potentiometer.*

Task 24. *Apply 30 mV peak-to-peak signal as the input and observe the output waveform.*

Task 25. *Reduce the value of R_1 until the output is distorted.*

Task 26. *Draw the output signal in the space provided.*

Task 27. *Increase the value of R_1 until the output is once again distorted.*

Task 28. *Draw the output signal in the space provided.*

Task 29. *Comment on the effect of R_1 on the output voltage waveform.*

♣ The End ♣

Part IV

Telecommunication

Workshop 4: Communication Networks and Protocols

Objective: To observe and analyze the basic operation of a communications network using packet sniffing.

Outcome: After successfully completion of this session, the student would be able to

1. Become familiar with the Wireshark packet sniffing tool
2. Be able to appreciate how the layered protocol stack facilitates the functioning of a communications network.

Equipment Required:

1. A Personal Computer with a network interface
2. Internet connectivity
3. Wireshark (the most recent version)
4.

Components Required:

None

This lab is adapted from a series of `Wireshark` Labs provided as a supplement to *Computer Networking: A Top-Down Approach*, 7th ed., J.F. Kurose and K.W. Ross.

4.1 Introduction

Our understanding of communication networks and protocols can often be greatly deepened by "seeing protocols in action" and by "playing around with protocols". In this lab, we will be working with a "real" network environment, and use `Wireshark` to capture data packets that are passing through. You will run various network applications in different scenarios using your own computer and observe the network protocols "in action," interacting and exchanging messages with entities elsewhere in the Internet. Thus, you and your computer will be an integral part of these "live" labs. You'll observe, and you'll learn, by doing.

In this lab, you'll get acquainted with `Wireshark`, make some simple packet captures with it, and gain some insight into how communications networks operate.

4.2 Pre-Lab

Step 1: Please read Annex 4.5: "An Introduction to `Wireshark`" and install the software on your computer using the given instructions.
Step 2: Take `Wireshark` for a Test Run following the instructions given in Annex 4.5: "Wireshark Test Run".
Step 3:

Task 1. *Copy the protocol listing obtained after the last step in the Test Run in the space given in the Task Sheet.*

4.3 Observing the TCP/IP protocol stack

In this lab, you will observe the TCP/IP protocol in action using the common data communication activity of browsing the web. You will also learn some more features of `Wireshark` during this activity. Follow the steps given below.
Step 1: Erase your recent browsing history, execute the Test Run again and answer the following questions. Explain how you obtained your answer in each case.

Task 2. *How long did it take from when each HTTP GET message was sent until the corresponding HTTP OK reply was received? (By default, the value of the Time column in the packet-listing window is the amount of time, in seconds, since* `Wireshark` *tracing began. To display the Time field in time-of-day format, select the* `Wireshark` *View pull down menu, then select Time Display Format, then select Time-of-day.).*

Task 3. *What is the IP address of the gaia.cs.umass.edu (also known as www-net.cs.umass.edu)? What is the IP of your computer?*

Step 2: Select the first HTTP GET message and from the top menu select

Analyze -> Conversation Filter -> TCP.

Task 4. *Sketch the sequence of activities that you see between your computer and the web server at gaia.cs.umass.edu. The Conversation Filter extracts the entire sequence of messages (involving TCP in this case) between the endpoints in the selected message.*

Step 3: Remove the packet filtering.

Task 5. *List 3 different protocols that appear in the protocol column in the packet-listing window.*

Step 4: Using the filter, select only the messages involving your computer (either as source or destination).

Task 6. *What new destinations do you observe? What new protocols do you see in action? What are their purposes?*

4.4 Discussion

Task 7. *Describe how this lab has helped sharpen your interest in communication networks and protocols. Also state how this lab can be improved in future. Use 150 – 300 words.*

4.5 Annex: An introduction to Wireshark

4.5.1 What is a Packet Sniffer ?

The basic tool for observing the messages exchanged between executing protocol entities is called a **packet sniffer**. As the name suggests, a packet sniffer captures ("sniffs") messages being sent/received from/by your computer; it will also typically store and/or display the contents of the various protocol fields in these captured messages. A packet sniffer itself is passive. It observes messages being sent and received by applications and protocols running on your computer, but never sends packets itself. Similarly, received packets are never explicitly addressed to the packet sniffer. Instead, a packet sniffer receives a *copy* of packets that are sent/received from/by application and protocols executing on your machine.

Figure 4.1 shows the structure of a packet sniffer. At the right of Figure 4.1 are the protocols (in this case, Internet protocols) and applications (such as a web browser or ftp client) that normally run on your computer. The packet sniffer, shown within the dashed rectangle in Figure 4.1 is an addition to the usual software in your computer, and consists of two parts. The **packet capture library** receives a copy of every link-layer frame that is sent from or received by your computer. Messages exchanged by higher layer protocols such as HTTP (that we use to browse the web) all are eventually encapsulated in link-layer frames that are transmitted over physical media such as an Ethernet cable or a wireless medium. In Figure 4.1, the assumed physical media is an Ethernet, and so all upper-layer protocols are eventually encapsulated within an Ethernet frame. Capturing all link-layer frames thus gives you all messages sent/received from/by all protocols and applications executing in your computer.

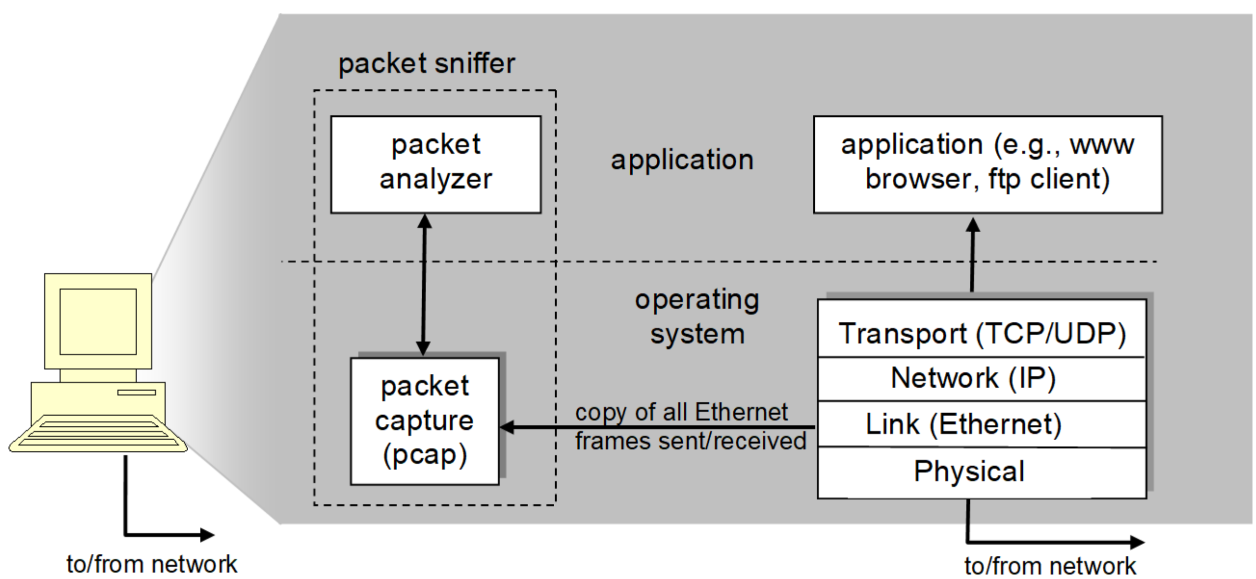

Figure 4.1: Packet Sniffer Structure

The second component of a packet sniffer is the **packet analyzer**, which displays the contents of all fields within a protocol message. In order to do so, the packet analyzer must "understand" the structure of all messages exchanged by protocols. For example, suppose we are interested in displaying the various fields in messages exchanged by the HTTP protocol in Figure 4.1. The packet analyzer understands the format of Ethernet frames, and so can identify the IP datagram within an Ethernet frame. It also understands the IP datagram format, so that it can extract the TCP segment within the IP datagram. Finally, it understands the TCP segment structure, so it can extract the HTTP message contained in the TCP segment. Finally, it understands the HTTP protocol and so, is able to identify the contents of HTTP message.

4.5.2 Wireshark

`Wireshark` is a packet sniffer http://www.wireshark.org/ which allows us to display the contents of messages being sent/received from/by protocols at different levels of the protocol stack. (Technically speaking, `Wireshark` is a packet analyzer that uses a packet capture library in your computer). `Wireshark` is a free network protocol analyzer that runs on Windows, Mac, and Linux/Unix computer. It's an ideal packet analyzer for our labs – it is stable, has a large user base and well-documented support that includes a user-guide (http://www.wireshark.org/docs/wsug_html_chunked/), man pages (http://www.wireshark.org/docs/man-pages/), and a detailed FAQ (http://www.wireshark.org/faq.html), rich functionality that includes the capability to analyze hundreds of protocols, and a well-designed user interface.

4.5.3 Installing Wireshark

In order to run `Wireshark`, you will need to have access to a computer that supports both `Wireshark` and the *libpcap* or *WinPCap* packet capture library. The *libpcap* software will be installed for you, if it is not installed within your operating system, when you install `Wireshark`. See http://www.wireshark.org/download.html for a list of supported operating systems and download sites

Download and install the `Wireshark` software:

- Go to http://www.wireshark.org/download.html and download and install the `Wireshark` binary for your computer.

4.5.4 Running Wireshark

When you run the `Wireshark` program, you'll get a startup screen that looks similar to Figure 4.2. (Note: Different versions of `Wireshark` will have different startup screens.) In the Capture section of the screen, there is a list of communication interfaces available in the computer. To capture packets, you need to first select an interface from this list.

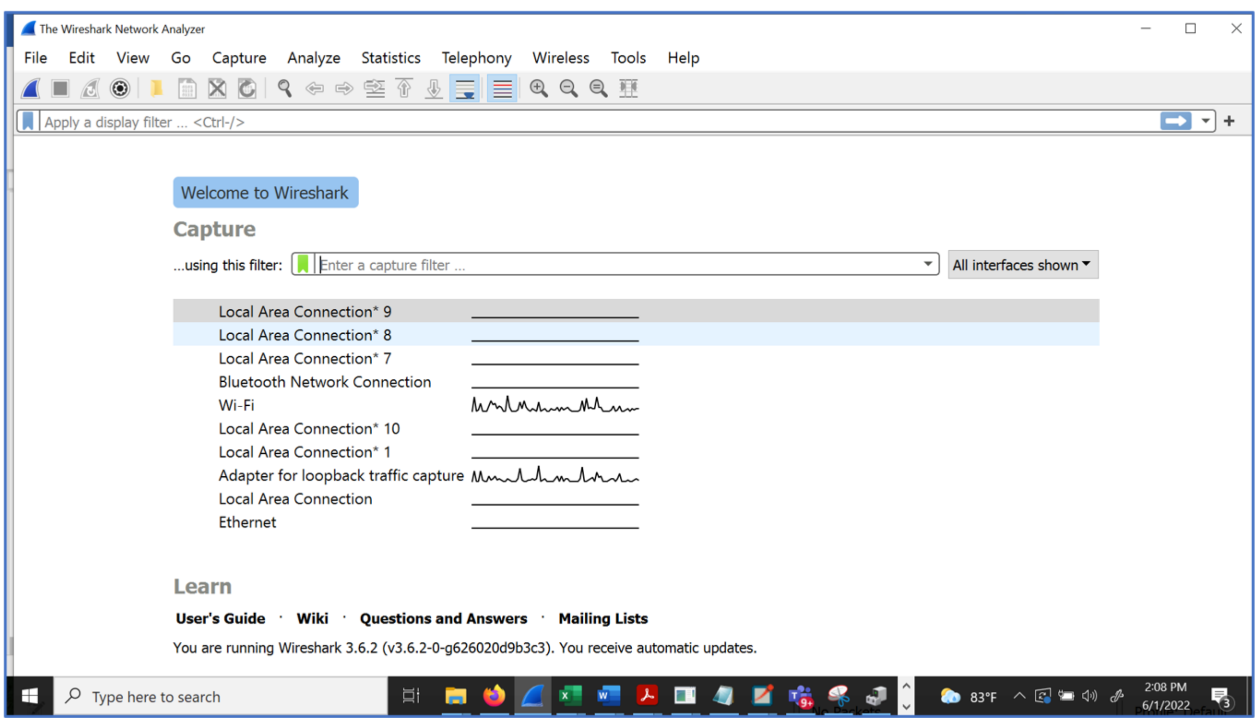

Figure 4.2: The Wireshark start-up screen

When you click on one of these interfaces to start packet capture (i.e., for `Wireshark` to begin capturing all packets being sent to/from that interface), a screen like the one in Fig. A16.1.3 will be displayed, showing information about the packets being captured. Once you start packet capture, you can stop it by using the *Capture* pull down menu and selecting *Stop*.

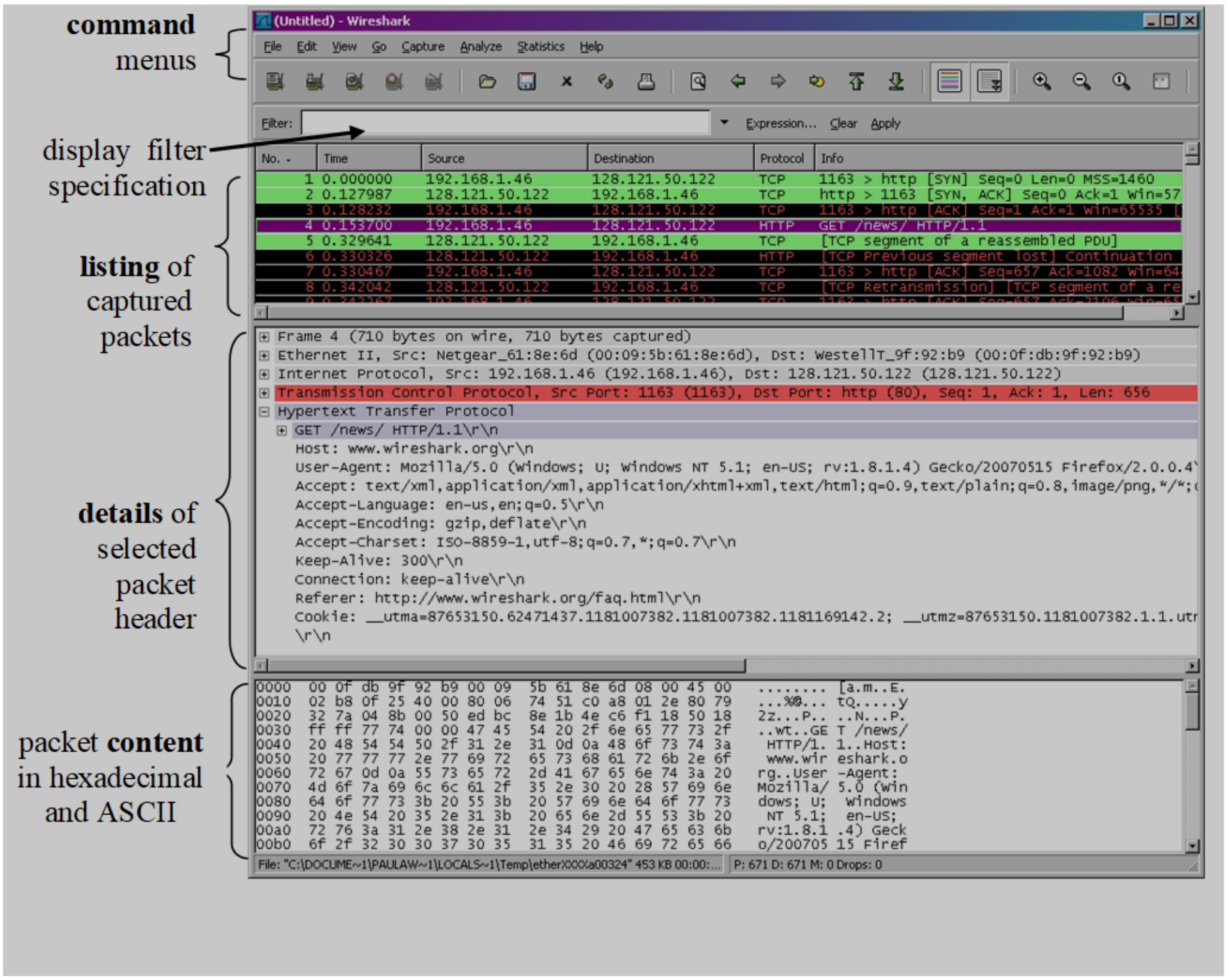

Figure 4.3: Wireshark Graphical User Interface during packet capture and analysis

The `Wireshark` interface has five major components as shown in Figure 4.3. These are described below:

- The **command menus** are standard pulldown menus located at the top of the window. Of interest to us now are the File and Capture menus. The File menu allows you to save captured packet data or open a file containing previously captured packet data, and exit the `Wireshark` application. The Capture menu allows you to begin packet capture.

- The **packet-listing window** displays a one-line summary for each packet captured, including the packet number (assigned by `Wireshark`; this is *not* a packet number contained in any protocol's header), the time at which the packet was captured, the packet's source and destination addresses, the protocol type, and protocol-specific information contained in the packet. The packet listing can be sorted according to any of these categories by clicking on a column name. The protocol type field lists the highest-level protocol that sent or received this packet, i.e., the protocol that is the source or ultimate sink for this packet.

- The **packet-header details window** provides details about the packet selected (highlighted) in the packet-listing window. (To select a packet in the packet-listing window, place the cursor over the packet's one-line summary in the packet-listing window and click with the left mouse button.) These details include information about all the protocols over which the packet has been carried, up to the highest level.

- The **packet-contents** window displays the entire contents of the captured frame, in both ASCII and hexadecimal format.
- Towards the top of the `Wireshark` graphical user interface, is the **packet display filter field**, into which a protocol name or other information can be entered in order to filter the information displayed in the packet-listing window (e.g. to select and display only packets using a particular protocol).

4.6 Annex: A Wireshark Test Run

Follow the steps given below to complete your first run with `Wireshark`. You will access a web page, and examine the protocols that enable you to do so.

1. Start up your web browser, which will display your selected homepage.
2. Start `Wireshark`.
3. To begin packet capture, select the Capture pull down menu and select *Options*. This will cause the "Wireshark: Capture Options" window to be displayed with the available interfaces as shown in Figure 4.1 earlier.
4. Double-click on the interface that you wish to capture packets on (i.e., the network interface you use for Internet access). Packet capture will now begin - `Wireshark` is now capturing all packets being sent/received from/by your computer! Once you begin packet capture, a window similar to that shown in Figure 4.3 will appear. This window shows the packets being captured. By selecting `Capture` pulldown menu and selecting *Stop*, you can stop packet capture. But don't stop packet capture yet.
5. Let's capture some interesting packets now. To do so, we'll need to generate some network traffic. We will look at the HTTP protocol that is used to download content from a website. While `Wireshark` is running, enter the URL: `http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html` and have that page displayed in your browser.

   In order to display this page, your browser will contact the HTTP server at gaia.cs.umass.edu and exchange HTTP messages with the server in order to download this page. The Ethernet frames containing these HTTP messages (as well as all other frames passing through your Ethernet adapter) will be captured by `Wireshark`.
6. After your browser has displayed the INTRO-wireshark-file1.html page (it is a simple one line of congratulations), stop `Wireshark` packet capture. The main window should now look similar to Figure 4.3.
7. You now have live packet data that contains all protocol messages exchanged between your computer and other network entities! The HTTP message exchanges with the gaia.cs.umass.edu web server should appear somewhere in the listing of packets captured. But there will be many other types of packets displayed as well (see, e.g., the many different protocol types shown in the *Protocol* column in Figure 4.3). Even though the only action you took was to download a web page, there were evidently many other protocols running on your computer that are unseen by the user. We'll learn much more about these protocols as we progress through the text! For now, you should just be aware that there is often much more going on than "meet's the eye"!
8. Type in "http" (without the quotes, and in lower case – all protocol names are in lower case in `Wireshark`) into the display filter specification window at the top of the main `Wireshark` window. Then select *Apply* (to the right of where you entered "http"). This will cause only HTTP message to be displayed in the packet-listing window.

♣ The End ♣