

# CS 2600 Packet Sniffer Lab

## Using Wireshark v3.2 and Npcap v0.99 or libpcap

### Lab Exercise 3: Wireshark Installation and Introduction

Approximate time required: 15 minutes (installation), 1 hour (lab)

Copyright © 2020 David Heldenbrand. All rights reserved.

#### Introduction

Wireshark is a network analyzer (“packet sniffer”) created by the open source community. A packet sniffer can selectively capture, parse and analyze network traffic. There are versions of Wireshark for Linux, Microsoft Windows and Apple macOS. This lab will assume the Microsoft Windows platform, but you are welcome use Linux or macOS for the Wireshark lab exercises. The Wireshark user interface is fairly consistent across all three operating systems, and no adjustments to the lab exercise will be required, except as noted. Wireshark will run on a VMware virtual machine, but that configuration complicates some lab exercises, and is not recommended if you have the option of running it directly on the host operating system.

You can access the Wireshark User’s Guide from within Wireshark using **Help | Contents**. The equivalent User’s Guide called is available at <https://www.wireshark.org/docs/> in several formats.

Other useful references include the following:

- Wireshark Network Analysis (Second Edition): The Official Wireshark Certified Network Analyst Study Guide by Laura Chappell (Laura Chappell University publ., \$99.95, ISBN-13 978-1893939943)
- Wireshark 101: Essential Skills for Network Analysis by Laura Chappell (Laura Chappell University publ., \$49.95, ISBN-13 978-1893939721)
- Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems, 2<sup>nd</sup> Ed. by Chris Saunders (No Starch Press publ., \$49.95, ISBN-13 978-1-59327-149-7)

#### Installing Wireshark

The Wireshark labs in this course can be completed in the Network Lab in CS 516 or on your own Microsoft Windows (v7, 8.1 or 10), macOS (10.5 and later) or Linux system. If you use your own computer, you will need Internet access via Ethernet or WiFi. Wireshark is much easier to install and manage than Riverbed Modeler, and we recommend that you install it on your own computer if you have the option.

Because of potential security and privacy issues, obtain permission from your network administrator before running Wireshark at work. If you are doing this lab exercise on a computer in the Network Lab (instead of installing it on your own computer), skip to the section titled **Testing the Installation**, and take the textbook or a copy of the Lecture 18 slides with you to the lab for reference.

#### The Npcap and libcap Packet Drivers

As of Wireshark v3.0.0, the Windows version of Wireshark ships with a special packet-capturing driver called Npcap. Linux and macOS-based systems require libcap instead, which is pre-installed in macOS and in most Linux distributions.

Wireshark, Npcap and libcap are open source software available free of charge. A few WiFi adapters may not be supported by Npcap, and some have limited capture capabilities. Support information for specific adapters is available at [https://secwiki.Org/w/Npcap/WiFi\\_adapters](https://secwiki.Org/w/Npcap/WiFi_adapters).

## Installing Wireshark and Npcap on Windows

(Skip ahead for macOS and Linux installation.)

All Windows drivers have the potential to disrupt a Windows configuration, so create a Windows restore point before running the Wireshark installation on Windows. *If you have an older version of Npcap or WinPcap already installed, uninstall it manually before you attempt to install Wireshark v3.2.x.*

The Wireshark installation process may reboot your computer, so close other programs before you begin. Browse to <https://www.wireshark.org/download.html> to obtain the latest stable release of the Wireshark Windows installer. Unless you have a very old computer, download the “Windows Installer (64-bit)” version. New releases of Wireshark are issued frequently. We will stay with the current version for the rest of the semester.

Make sure that you don’t have an old version of Wireshark running and then run the installer with administrative privileges. You can take the installation defaults as desired. You do not need to install USBPcap. Click **Install** at the bottom. Accept the Npcap license agreement, take the default installation options for Npcap, and then click **Next** and **Finish** for this embedded Npcap install. (Npcap can take a couple of minutes to install.) When the main Wireshark installation completes, click **Next** and **Finish**.

## Installing Wireshark on macOS

The Wireshark installation process may reboot your computer, so close other programs before you begin. Browse to <https://www.wireshark.org/download.html> to obtain the latest stable 64-bit release of the Wireshark macOS installer. The current Wireshark 3.0 macOS package requires macOS version 10.12 (Sierra) or later. If you’re running an older version of macOS, use Wireshark 2.6. New releases of Wireshark are issued frequently. We will stay with the current version for the rest of the semester.

Open the installer with DiskImageMounter. After downloading the Wireshark package, double click it and follow the directions and take the defaults to run the installation.

## Installing Wireshark on Linux

Check to see if Wireshark was provided with the Linux distribution that you are running. If not, check the **Third-Party Packages** section at the bottom of <https://www.wireshark.org/download.html> to obtain the latest version for your distribution. The Wireshark installation process may reboot your computer, so close other programs before you begin.

## Testing the Installation

Now we will verify connectivity between the computer running Wireshark and the “target” computer that you will be communicating with (which can be on your home network, in the Network Lab or out on the Internet). If you will be using one of your own machines as the target computer, boot it. If you have a choice between Ethernet and WiFi, use your target computer’s Ethernet connection. If not, WiFi will work.

Type `ipconfig` in the target computer’s Windows command line window to obtain its IPv4 address. (Use `ifconfig` in a Linux or macOS terminal window.) If the target has multiple network adapters, be sure to read the IPv4 address of the appropriate adapter. (With macOS, this will be labeled “inet”, probably on `en0`. On Windows, this will *not* be the Npcap Loopback adapter, and the IPv4 address will not begin with 169...)

If you are targeting a computer out on the Internet (because you don’t have a second computer available) or if you are working in the Network Lab (CS 516) , use `ietf.org` or `acme.com` as your target.

If you are running a firewall on your own computer(s) that filters ICMP Echo Request and/or Reply messages (commonly called “ping”), disable that filter for the duration of this lab. See [https://technet.microsoft.com/en-us/library/cc749323\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc749323(v=ws.10).aspx) for instructions. Then open a command line or terminal window on the computer running Wireshark and type...

```
ping <IP address/domain name>
```

.replacing the bracketed field with either the domain name or IP address of the other computer, for example:

```
ping ietf.org
or
ping 4.31.198.44
```

Use CTRL+C to stop the pings in macOS and Linux.

You should see multiple replies being returned which look very similar to this (numbers will vary):

```
Reply from 4.31.198.44: bytes=32 time=71ms TTL=56
```

If you receive a reply with a longer hexadecimal address, like this.

```
Reply from fe80::dcec:2db6:1c32:522a: time100ms
```

...your ping used an IPv6 address. (This often occurs when your ISP is Comcast/Xfinity.)

We need to use IPv4 here, so retry the ping using this syntax to force IPv4 (Windows only):

```
ping -4 ietf.org
```

If you are using Windows, you will need to remember to add the -4 when you use ping in future lab exercises. Or you can temporarily disable IPv6 in **Control Panel | Network and Sharing Center**. Click **Change adapter settings**, right click on the adapter you are using, click **Properties**, and temporarily uncheck **Internet Protocol Version 6**.

If you see a ping error message, similar to one of these...

```
Request timed out
```

```
Reply from 4.31.198.44: Destination host unreachable
```

```
Ping request could not find host ietf.org. Please check the  
name and try again.
```

...this indicates a communication problem between the two computers. For security reasons, some versions of the Windows Firewall block ping (ICMP Echo Request and Reply) by default. And some commercial web servers have firewalls that block pings. So you may need to try different target computer until you find one that responds. (You could try your home router, which is often configured with 192.168.1.1 as the IPv4 address.) Resolve this problem before continuing, or else complete the lab exercise in the Network Lab.

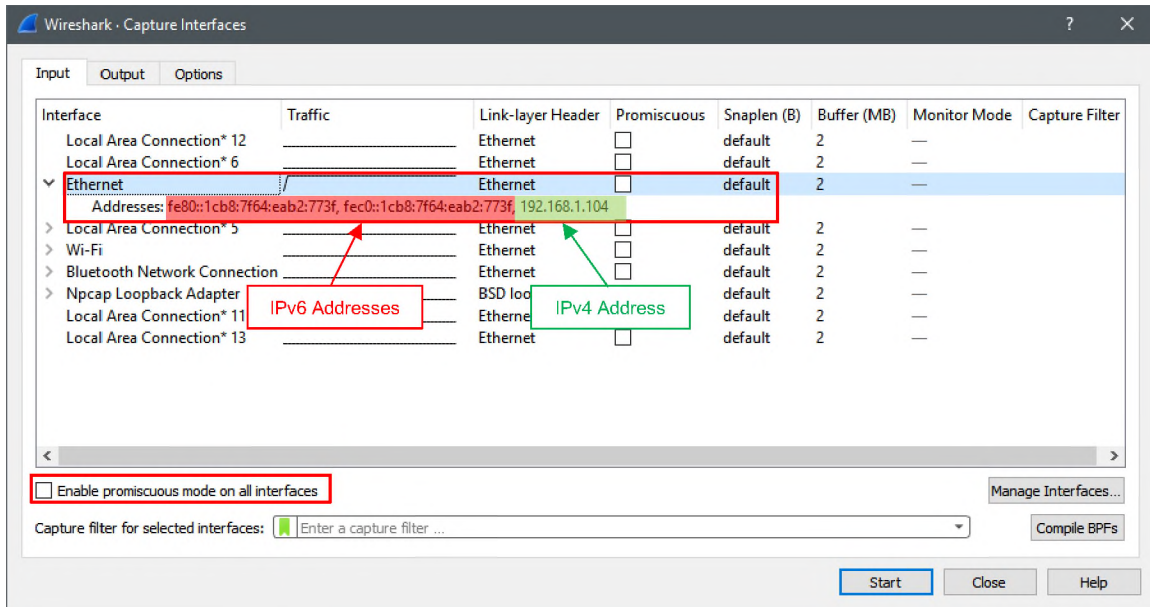
## Running Wireshark

After you have used ping to confirm connectivity with the target computer, start Wireshark. Administrator privileges are not normally required, but if you have problems getting Wireshark to run, or with printing from Wireshark, you may need to use an Administrative login. (The login in the Network Lab has administrator privileges.)

Click **Capture | Options...** In the *Capture Interfaces* dialog box, you will see several interfaces ("WiFi", "Ethernet", "Local Area Connection", etc.). The active interface that you used to ping the target computer will be displaying a miniature jagged scrolling chart of packet traffic levels. You may also see traffic on interfaces labeled "Npcap" or "Adapter for loopback traffic capture". Those can be ignored.

Expand your active interface by clicking on the ">" to the left. You may see one or more hexadecimal IPv6 addresses, along with the dotted decimal IPv4 address associated with this interface. (See figure 1 below.)

(CONTINUED ON NEXT PAGE)



**Figure 1. Enabling the Active Interface (Windows Version Using Ethernet)**


If you have virtual machines installed, the virtual adapters will be listed as well, so use the scrollbar to view the entire list. If you are using WiFi and you see multiple “Wireless Connection” adapters listed, pick the adapter with the non-zero IPv4 address, often something like 192.168.x.x.

If you don’t see any interfaces listed on your Windows computer, it is likely that the Npcap driver has not loaded. If you are logged in with a User ID that doesn’t have administrator privileges, Npcap may not have loaded automatically at startup. (Administrative privileges are generally required to dynamically load Windows drivers.) You can troubleshoot this problem by logging in with administrator privileges and repeating the steps in this section. If that test is successful, you may wish to uninstall and reinstall Wireshark, taking care to check the **Automatically start the Npcap driver at boot time** option during the installation process.

Select your active interface, as shown in figure 1.

The *Enable promiscuous mode on all interfaces* option (toward the lower left in the *Capture Interfaces* dialog box) allows Wireshark to capture network traffic that is visible on the selected interface, including packets not being sent to or from the computer running Wireshark. Few built-in WiFi adapters support promiscuous mode, and it isn’t generally useful on Ethernet unless you are using an old-style Ethernet hub for testing. If you are using a wireless adapter, you may need to uncheck the *Enable promiscuous mode on all interfaces* option to get the packet capture to work. Leave it unchecked for now.

Open a command line window (or a terminal in Linux or macOS). Back in the Wireshark *Capture Interfaces* dialog box, click **Start** to begin the packet capture.

To generate some packets for capture, `ping` the same target computer again from the command line window. In the upper Wireshark pane, you should see some ICMP (ping) packets being captured, possibly along with TCP, UDP, ICMPv6, DNS and ARP packets or other traffic. Verify that your pings are receiving replies in the command line or terminal window, then click the red “stop” icon  at the upper left.

## Analyzing the Captured Traffic

You should see a display of the captured traffic in the *Packet List* pane (see figure 2 below). Summary information for each packet is displayed on a single line. By default, Wireshark assigns different display colors to different packet types. Detailed information about the content of the currently selected packet is interpreted or “dissected” in the *Packet Details* pane in the middle. The “raw” hexadecimal and ASCII character content of that same packet (as it would appear in memory or “on the wire”) is shown in the *Packet Bytes* pane at the bottom. Hex values which don’t translate to an ASCII character are represented by a dot.

If the *Packet List*, *Packet Details* and/or *Packet Bytes* panes are not visible, use the *View* menu at the upper left to display them.

If no packets are displayed, it may be because you did not select an interface, or because you selected the wrong interface. For example, you may have selected your wireless adapter in Wireshark and then sent your ping messages over the Ethernet interface. Try selecting a different interface, restarting the Wireshark capture and pinging the target computer again until you see packets being captured in the *Packet List* pane.

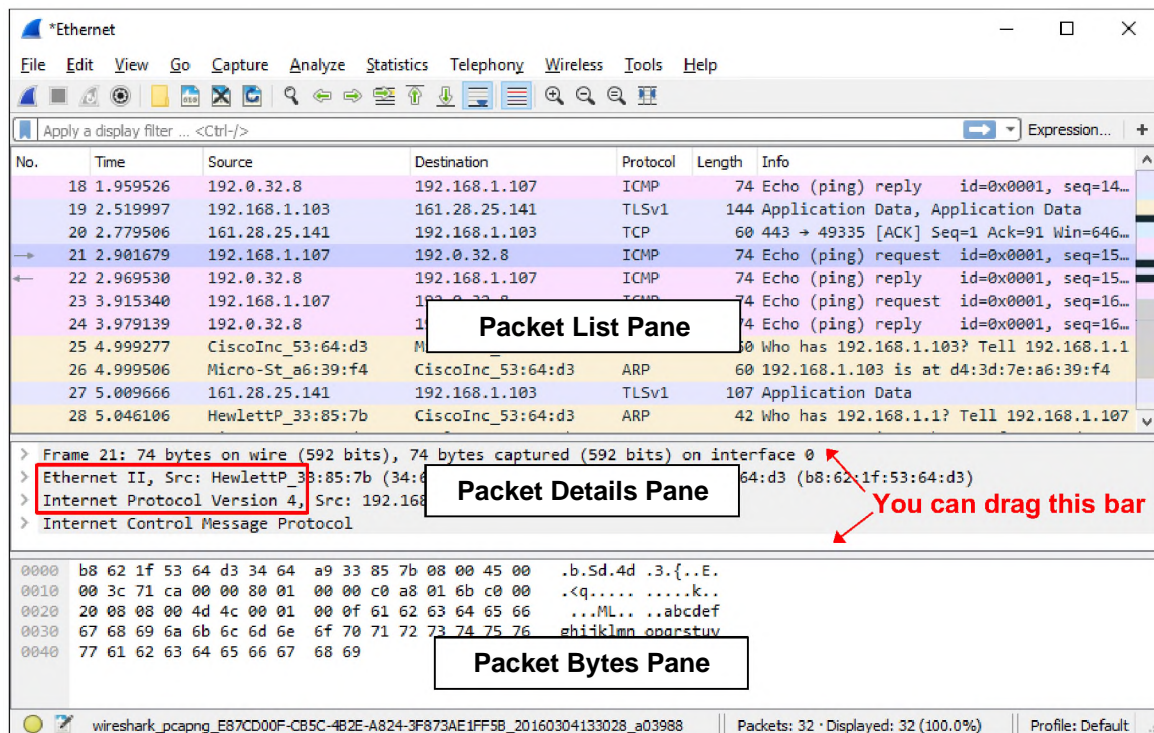


Figure 2. The Wireshark Main Screen (Windows Version)

You have captured a group of Ethernet or WiFi (802.11) *frames*, most of which will contain IP *packets*. Those IP packets may contain various payloads including ICMP (ping), TCP or UDP messages. The Wireshark interface and much industry documentation tend to use the terms *frame* and *packet* interchangeably. Strictly speaking, a *frame* is not a *packet*. Rather, a *packet* is carried within a *frame*. The frame header is the second line in the *Packet Details* pane (highlighted as “Ethernet II” in figure 2 above). It will be listed as “Ethernet II”, even if you are capturing WiFi frames. The packet header is the next line below, typically “Internet Protocol Version 4”.



In the (upper) *Packet List* pane, scroll to the top, if necessary, and select the first ICMP (not IGMP or ICMPv6) packet listed in the *Protocol* column. It should be labeled “Echo (ping) request” in the *Info* column at the right. (Ignore any other types of ICMP packets such as “Destination unreachable” etc., and don’t let the dark highlighting on other packet types confuse you.)

In the *Packet Details* pane below, click the “>” to the left of “Frame...” to see summary information about the frame. (This line does not represent a specific header. If you can’t see all of the expanded frame information, drag the bottom edge of the *Packet Details* pane down to enlarge that pane.) The elapsed time (delta) between frames is listed in seconds in the “Time” column of the *Packet List* pane above.

### Answer the following questions to complete the lab:

1. What is the frame length in bytes, as reported by Wireshark? (“Frame Length” in Wireshark is the overall frame length, not including the 8-byte frame preamble and 4-byte trailer (CRC), which are removed by the network adapter before the captured frame reaches Npcap or libcap.) Use figure 2.25 in the text or the corresponding Lecture 18 slide titled “Ethernet Type II Frame Format” as a reference. Click the “v” to the left of “Frame” to contract the frame summary information.

2. In the *Packet Details* pane, click the “>” to the left of “Ethernet II”. This will expand to show you the contents of the three Ethernet II header fields. (Refer to the Lecture 18 slide titled “Dissecting an Ethernet Type II Frame in Wireshark”.)

What are the Ethernet (not IPv4) Destination and Source MAC addresses? List them in the standard hexadecimal format used by Wireshark (xx:xx:xx:xx:xx:xx). Note that selecting the Destination or Source address line in this Ethernet header details section will highlight the address in the left-hand hex display in the *Packet Bytes* pane at the bottom.

If you are capturing WiFi traffic, note that Wireshark displays the 802.11 WiFi header as if it were an Ethernet header. In reality, there are major differences between 802.3 and 802.11 frame headers that Wireshark/Npcap on Windows ignores, because Windows blocks access to the 802.11 header. Those differences will be covered in the lectures on wireless LAN standards.

3. Who is the manufacturer of the Ethernet or WiFi adapter in the computer running Wireshark? (It may not be the same as the manufacturer of the computer.) Wireshark will usually indicate this as part of the Ethernet II information in the *Packet Details* pane. Look for something similar to AsustekC\_4c:d2:e3 (ASUS) or Hewlett-\_1f:0c:4d (Hewlett Packard). If you see the adapter manufacturer names, Wireshark is translating the manufacturer’s Organizational Unit Identifier in the first three bytes of the Ethernet header into an abbreviation of the manufacturer’s name.

If Wireshark is unable to determine the manufacturer, go to <https://www.wireshark.org/tools/oui-lookup.html>, enter the three left hand colon-separated hex bytes of each Ethernet address in the “OUI search” box, and click **Find**. You’ll see the corresponding OUI at the bottom under “Results”. For background, refer to the Lecture 18 slide titled “IEEE Ethernet Vendor Organizational Unit Identifiers”.)

4. What is the numeric value of the Ethernet II Type code in the frame header (“Type:”)? (The “0x” notation indicates a hexadecimal number.) What type of packet (frame payload) does that number indicate? Select the Type code in the middle *Packet Details* pane to highlight the Type code in the hex dump down in the *Packet Bytes* pane. You should be able to see the Type code beginning at an offset of 12 (decimal) bytes. Note that the byte offsets in the left-hand column of the *Packet Bytes* pane are in hex, so “10” hex = “16” decimal, etc.

5. Back in the *Packet Details* pane, contract the Ethernet header details, then click the ">" to the left of "Internet Protocol..." (IP) and scroll down to view the individual IP header fields. Which version of IP is being used? Note the "0100 . . . ." representation, indicating that the version field is only 4 bits long. What is the length of this IP header, in bytes?

6. What are the corresponding Internet Protocol (IP) source and destination addresses?

7. In the *Packet Details* pane, scroll down to next top-level node, "Internet Control Message Protocol". This contains "data" (the payload of the IP packet). In this case, the packet payload is an ICMP message containing the ping request (instead of the usual TCP or UDP message). ICMP is the control and error notification protocol for the Internet (IP) layer, as well as the protocol that sends and receives ping (echo request and reply) messages. ICMP resides immediately above IP in the stack, but unlike TCP and UDP, ICMP is not a true Transport protocol.

The `ping` command that you executed earlier will have sent multiple ICMP Echo Requests and (hopefully) received ICMP Echo Replies. Inspect both types of ICMP messages by highlighting them, in turn, in the *Packet List* pane at the top. Note that there may be some unrelated TCP and/or UDP packets between an ICMP Echo Request and the corresponding Reply. What is the *ICMP Type* number for the Echo Request? What is the ICMP Type number for the Echo Reply?

8. While you are still displaying the ICMP Echo *Reply* information from the previous question, adjust the Wireshark display so that the expanded ICMP message is visible in the *Packet Details* pane, *make a screen shot of Wireshark, and paste it into your lab report following the answers to your questions.*

## Summary of Deliverables

- Answers to questions 1 - 7.
- Screenshot of ICMP Echo *Reply* Frame (question 8)

Submit your Lab Report to Canvas by the due date. If you are working in the lab, save your files (if you wish) and shut down your workstation.

If you are working on your own computer and have modified your firewall settings, *don't forget to restore them.* If you disabled IPv6, re-enable it (see pp. 3 - 4).