ITS 332 Networking Lab

Wireshark

Prepared by Dr Steven Gordon, 23 November 2007, Revision 50.

First Name: ____________ Last Name: ____________

Date: ________ ID: | | | | | | | | | |

1 Overview

This lab will introduce you to an application for capturing traffic on networks. By "capturing", we mean record and view the details of every packet sent and received by the computer. The application is called *Wireshark*[1]. Packet capture applications are useful to inspect the details of the network operations being performed by your computer (and the network), thereby used to diagnose problems. We will use it in the remaining labs to understand how protocols work.

2 Background

The implementation of protocol layers in a network device (computer, router, switch, etc.) is done in a mix of hardware and software. Typically the Physical and Data Link layer are implemented in hardware, e.g. on an Ethernet LAN card. *Drivers* are special pieces of software that provide an interface from the operating system to a specific hardware device. That is, the Ethernet driver provides the functions for your operating system to receive Ethernet frames (and put them into memory) from your LAN card. The operating system normally implements the Network and Transport layers in software: that is, there is a software process that implements IP, as well as separate processes to implement UDP, TCP, ICMP and other transport layer protocols. Finally, each individual application (like web browsers, email clients, instant messaging clients) implement the Application layer protocols (such as HTTP and SMTP), as well as the user functionality and interface specific to that application. Figure 1 illustrates the layers and their implementation.

When a signal is received by your LAN card the signal is processed by the Physical and Data Link layers, and an Ethernet frame is passed to the operating system (via the Ethernet network driver). Normally the operating system will process the frame, sending it to the IP software process, which eventually sends the data to the transport layer protocol software process, which finally sends the data to your application.

In order to view all the frames received by your computer, we use special *packet capture* software, that allows all the Data Link layer frames sent from LAN card to operating

[1] Previously it was called *Ethereal*

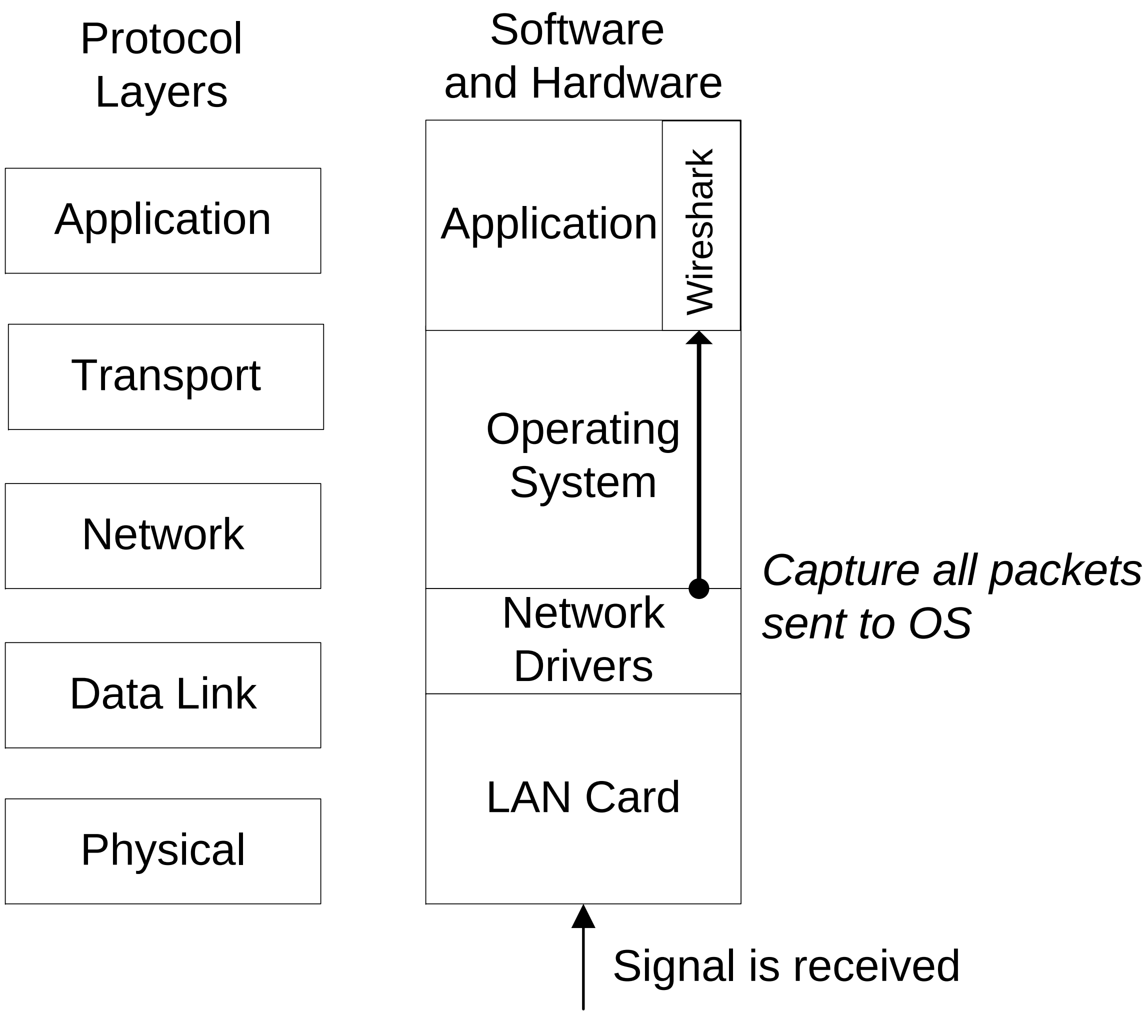

Figure 1: Capturing packets in the Operating System

system to be viewed by a normal application (in our case, Wireshark). The capturing of packets makes a copy of the exact packet receive by your computer—it does not modify the original packet. This allows us to analyse data received by the computer, in order to perform various network management tasks (such as diagnose problems, measure performance, identify security leaks).

3 Wireshark

3.1 Starting Wireshark

To start Wireshark, go to the menu: *Applications→Internet* and select *Wireshark (as root)*. (It is important you select *Wireshark (as root)* instead of just *Wireshark* because you will then have the privileges to capture all traffic).

If the menu item is not available, then possibly Wireshark is not installed. Let the Lab Instructor know if this is the case - the instructor will explain how to install Wireshark.

3.2 Capturing Traffic

The first thing to do in order to capture traffic is to select the interface to capture on. Remember your computer has several interfaces: two Ethernet interfaces and a loopback interface. Go to the *Capture→Interfaces...* menu item in Wireshark and select your interface. You want an Ethernet interface (e.g. `eth2` or `eth3`) - the one that is currently connected to the network. Select the interface and press *Start*. This will start capturing traffic. You will see a window displayed that counts the different types of packets captured. Press the *Stop* button to end the packet capture.

3.3 Viewing Captured Traffic

After a packet capture has finished, the main Wireshark window shows the captured packets. The window is split into three sections:

1. The top section (packet list) showing the list of capture packets. Each packet has the following information:
    - Packet number (with respect to the total number of packets captured)
    - Time the packet is captured, assuming the time the first packet captured is time 0.0
    - The source and destination IP addresses of the packet
    - The highest layer protocol associated with the packet
    - Summary information about the information carried by the packet
2. The middle section (packet details) showing detailed information about the packet selected in the top section. This is separated based on the layers of the packet.
3. The bottom section (packet bytes) showing the hexadecimal and ascii representations of the packet data.

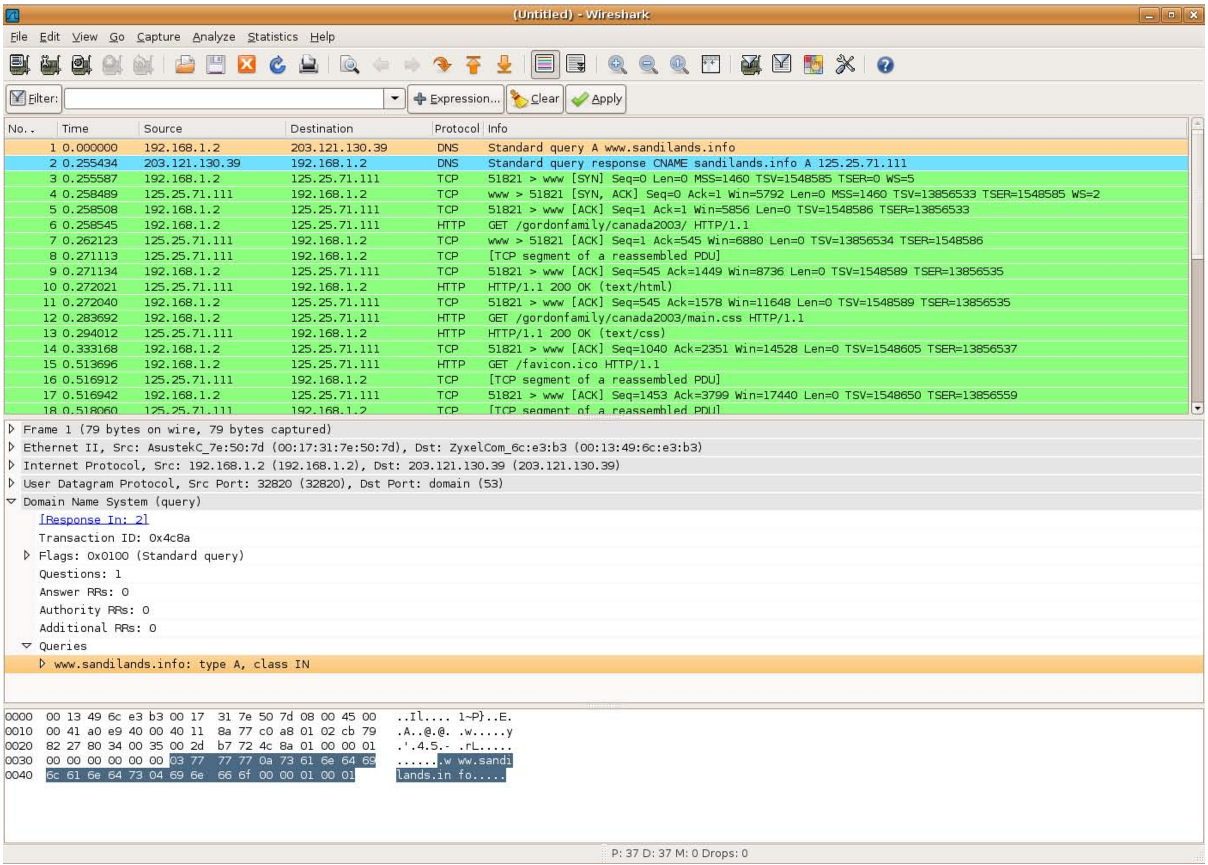

Figure 2: Main window of Wireshark

When selecting the first packet (in the top section), and then selecting the Domain Name System (in the middle section), the details of the packet are shown as below. It tells us that a DNS query request for `www.sandilands.info` has been sent from `192.168.1.2` to `203.121.130.39`. Other details of the packet (such as UDP, IP and Ethernet options) can be seen as well.

3.4 Analysis and Statistics

Wireshark has many in-built statistics that allow you to analyse the captured packets. This is very useful, especially if you have many packets captured (1000's to millions). You should explore (that is, view them and try to understand what they show) the following from the *Statistics* menu:

- Summary
- Protocol Hierarchy
- Conversations
- Flow Graph
- HTTP
- Packet Length
- TCP Stream Graph

3.5 Filters

The example used above was for a small trace of less than 100 packets captured over 10 seconds. When capturing over a long time period (and hence thousands or hundreds of thousands of packets), it is often desirable to investigate a selected portion of the packets (for example, packets between certain pairs of hosts, or using a particular protocol). Hence filters can be applied during the packet capture (such that only packets that meet the specified criteria are captured - called *capture filters*) or after the capture (such that analysis is only performed on packets that meet the specified criteria - called *display filters*). There is no reason for you to use capture filters, instead you can use display filters.

Display filters are used mainly to view certain types of packets. They make analyzing the data easier. One place you can enter a display filter is just above the top (packet list) section. You can either type in the filter and press Apply or create the filter using the Expression command. Some example filters include:

The following filter can be used to display only packets that have source or destination IP address of `10.10.1.171`

```
ip.addr==10.10.1.171
```

The next filter can be used to display only packets that have IP address of `10.10.1.127` and do not have a TCP port address of 8080.

```
ip.addr==10.10.1.127 && !tcp.port==8080
```

The next filter displays only ICMP packets.

```
icmp
```

The next filter displays only packets exchanged with a web server (assuming the web server is using port 80).

```
tcp.port==80
```

Further details of the display filter language and where it can be applied can be found in the Wireshark manual.

4 Tasks

For all of the following tasks, make sure you have no applications running generating traffic on the network. That is, exit any instant messaging applications and close all web browsers that are not needed for the task.

For each task you should draw a diagram illustrating the exchange of packets observed in Wireshark, as well as answer the questions. As an example, a packet exchange should be illustrated as in Figure 3. A packet (including headers) should illustrate the headers at the Data Link layer and above (Network, Transport etc.). As an example, a packet should be illustrated as in Figure 4.

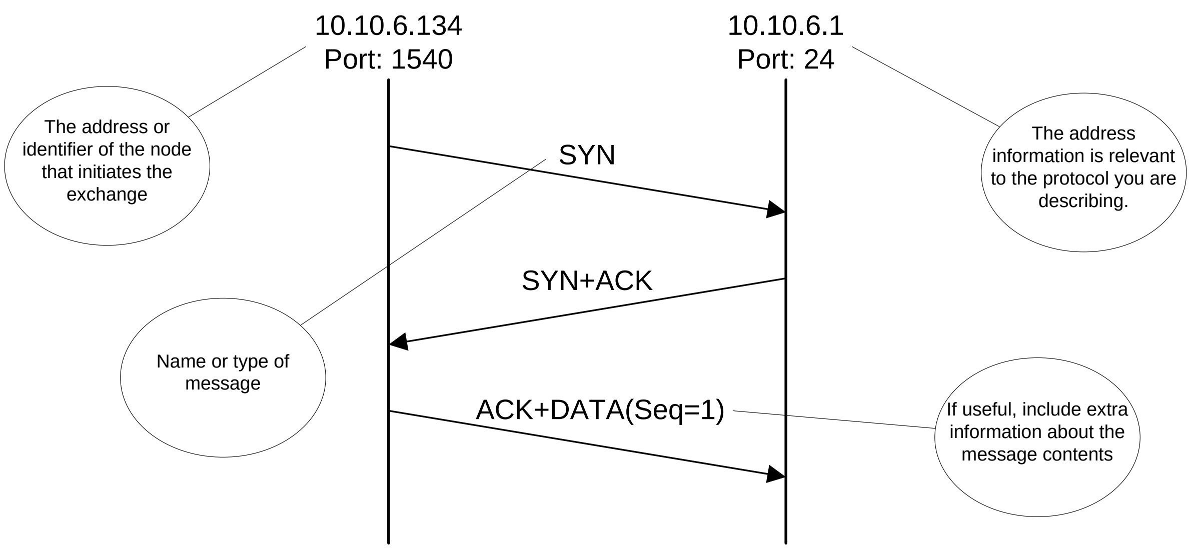

Figure 3: Example illustrating an exchange of packets

IEEE 802.3	IP	TCP	HTTP	Data

Figure 4: Example illustrating a packet

4.1 Ping

Task 1. *Capture traffic of a ping from your computer to another computer on the network. Limit the ping to send only 3 requests. Draw the exchange of packets.*

Task 2. *What is the default interval between ping requests?*

Task 3. *Draw a ping request packet.*

Task 4. *What does the first byte in the ICMP header indicate? What are the possible values for the first byte in the ping messages you captured, and what do they indicate?*

Task 5. *What is the default amount of data sent in the ICMP packet? Do you think the results of ping will change if the data size was larger? If so, how (explain your answer)?*

Task 6. *What is the interval between each ping request and ping reply as recorded by Wireshark? How does this compare to the results shown by ping? If there is a difference, explain a possible reason.*

4.2 Web Access

Before performing the following tasks, make sure you delete all cached information in your browser (in Firefox, go to *Tools→Clear Private Data*).

You will access files on the Lab server (the instructor will inform you of the IP address of this computer) which are specific for each student. The files are based on your ID number. For example, if your ID is 4812345678, then the files are:

- $/ITS332/student-files/4812345678/4812345678test.html$ - test HTML page.
- $/ITS332/student-files/4812345678/4812345678medium.bin$ - medium binary file (about 100KB).
- $/ITS332/student-files/4812345678/4812345678large.bin$ - large binary file (about 10MB).

Task 7. *Capture the traffic when you access your test HTML page on the Lab server. Draw the exchange of packets.*

Task 8. *View the Flow Graph for the exchange of packets (make sure you select* TCP Flow*, not* General Flow*). Check whether your diagram from Task 7 is correct.*

Task 9. *Record the following details: port number used by your web browser; size TCP data sent to web server; size of TCP data sent from web server to browser; the window advertised by the web server when sending the HTML page to the browser.*

Task 10. *Use the* Follow TCP Stream *analysis method to view the details of the HTTP request and response. Try to identify what the options in the HTTP request and response are used for.*

4.3 Web File Download

Again, your web browser is used (so clear the cache), but this time to download a binary file: your large binary file.

Task 11. *Capture the traffic when you access your large binary file on the Lab server. You do not have to draw the exchange of packets.*

Task 12. *View the* Protocol Hierarchy *statistics. What is the TCP throughput (Mb/s) measured? What is the total number of bytes transferred? What is the size of the large binary file? What is the efficiency of the data file download?*

Task 13. *View the* Packet Length *statistics. What are the ranges of the two most common packet lengths? What do you think these two ranges represent (that is, what type of packets)?*

Task 14. *What is the most common size of data sent in a TCP segment? Give a reason why the segment may be limited to this size.*

A Notes

Record any additional notes from this lab here (e.g. important points made by the instructor, summary of things you learned, mistakes you made). You should use this in future labs, as well as in preparation for assessment items like exams.