EEP-703 Computer Network Lab

Assignment 9-Capturing and analyzing network traffic using Wireshark Tool

Harshit Kumar Gupta
2013EET2369

Computer Technology
Department Of Electrical Engineering
IIT DELHI

April 15, 2014

Contents

Chapter 1

PROBLEM STATEMENT

Networks need watching, or network administration, observe network traffic on a realtime basis using (also called sniffing) protocol analysis tools such as wireshark. Now, you are given the following tasks:

1. List the different protocols you encounter when you visit a page (www.example.com) and explain their significance. Every URL you visit is hosted on a certain server having an IP address. Find out the IP addresses corresponding to URL you visited and your machine.

2. Open the first packet which has the IP addresses of www.example.com in the destination and has HTTP as the protocol.

   (a) Explain the five major headings: Frame, Ethernet Protocol, IPv4, TCP and HTTP. Why these different protocols are involved in the same message? How are these protocols related?

   (b) Using wireshark, observe normal network traffic. Can you separately see TCP/IP traffic, IPX traffic and NETBEUI traffic? What is the meaning of these different types of traffic?

   (c) Does the information in this packet state about the browser and OS you are using? Does it show that you are sending a cookie?

Chapter 2

ABSTRACT

1. The whole problem is to understand the Wireshark tools
2. Main Objective is to understand working of varios protocols.
3. Learnig how to capture network traffic.

Chapter 3

INTRODUCTION

Wireshark is a free and open-source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education. Wireshark is cross-platform, using the GTK+ widget toolkit in current releases, and Qt in the development version, to implement its user interface, and using pcap to capture packets.

Wireshark is very similar to tcpdump, but has a graphical front-end, plus some integrated sorting and filtering options.

Wireshark allows the user to put network interface controllers that support promiscuous mode into that mode, in order to see all traffic visible on that interface, not just traffic addressed to one of the interface's configured addresses and broadcast/multicast traffic.

Chapter 4

SPECIFICATIONS AND ASSUMPTIONS

Specifications

1. Data can be captured "from the wire" from a live network connection or read from a file of already-captured packets..
2. Live data can be read from a number of types of network, including Ethernet, IEEE 802.11, PPP, and loopback.
3. Captured network data can be browsed via a GUI, or via the terminal (command line) version of the utility, TShark

Assumptions

1. Data display can be refined using a display filter.
2. Wireshark put wireless network interface controllers into monitor mode.

Chapter 5

LOGIC USED/METHODOLOGY

The methodology that is used for developing this project work is defined below:

1. After downloading and installing Wireshark, you can launch it and click the name of an interface under Interface List to start capturing packets on that interface. For example, if you want to capture traffic on the wireless network, click your wireless interface. You can configure advanced features by clicking Capture Options, but this isn't necessary for now.

2. As soon as you click the interface's name, you'll see the packets start to appear in real time. Wireshark captures each packet sent to or from your system. If you're capturing on a wireless interface and have promiscuous mode enabled in your capture options, you'll also see other the other packets on the network.

3. Click the stop capture button near the top left corner of the window when you want to stop capturing traffic.

Chapter 6

EXECUTION DIRECTIVES

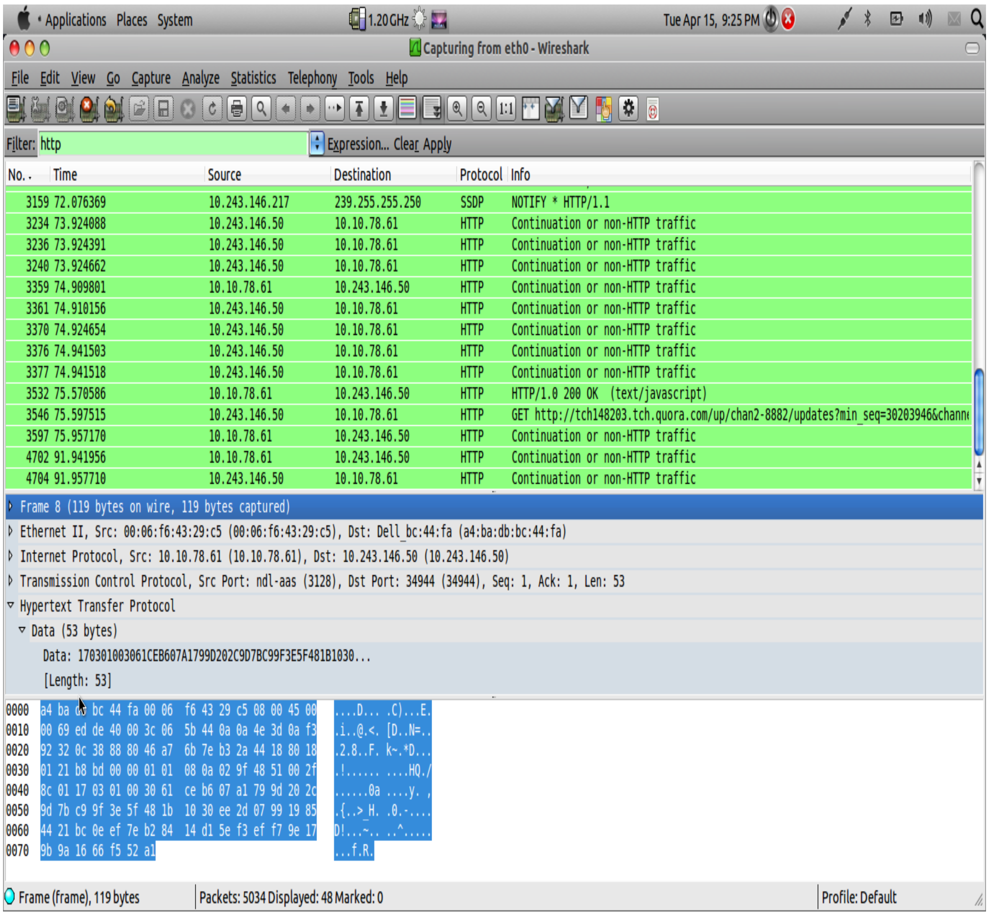
1. Filtering Packets—

   If you're trying to inspect something specific, such as the traffic a program sends when phoning home, it helps to close down all other applications using the network so you can narrow down the traffic. Still, you'll likely have a large amount of packets to sift through. That's where Wireshark's filters come in.

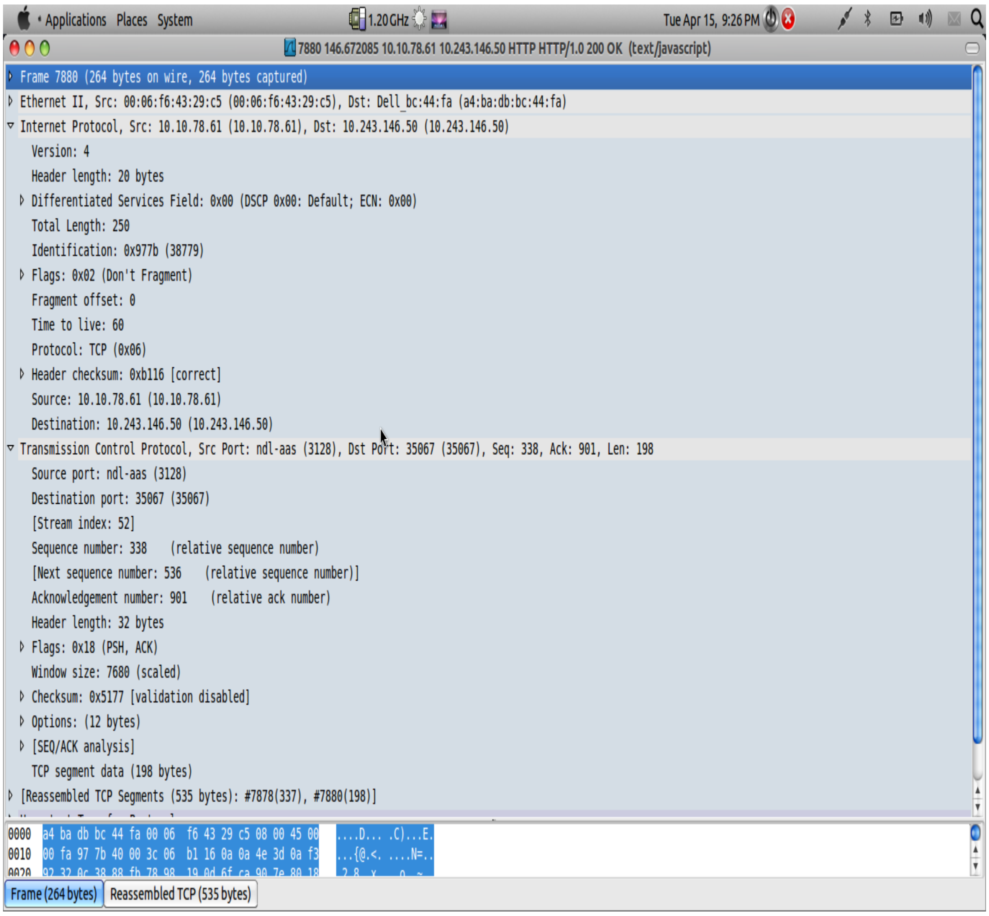
2. The most basic way to apply a filter is by typing it into the filter box at the top of the window and clicking Apply (or pressing Enter). For example, type "dns" and you'll see only DNS packets. When you start typing, Wireshark will help you autocomplete your filter.

Chapter 7

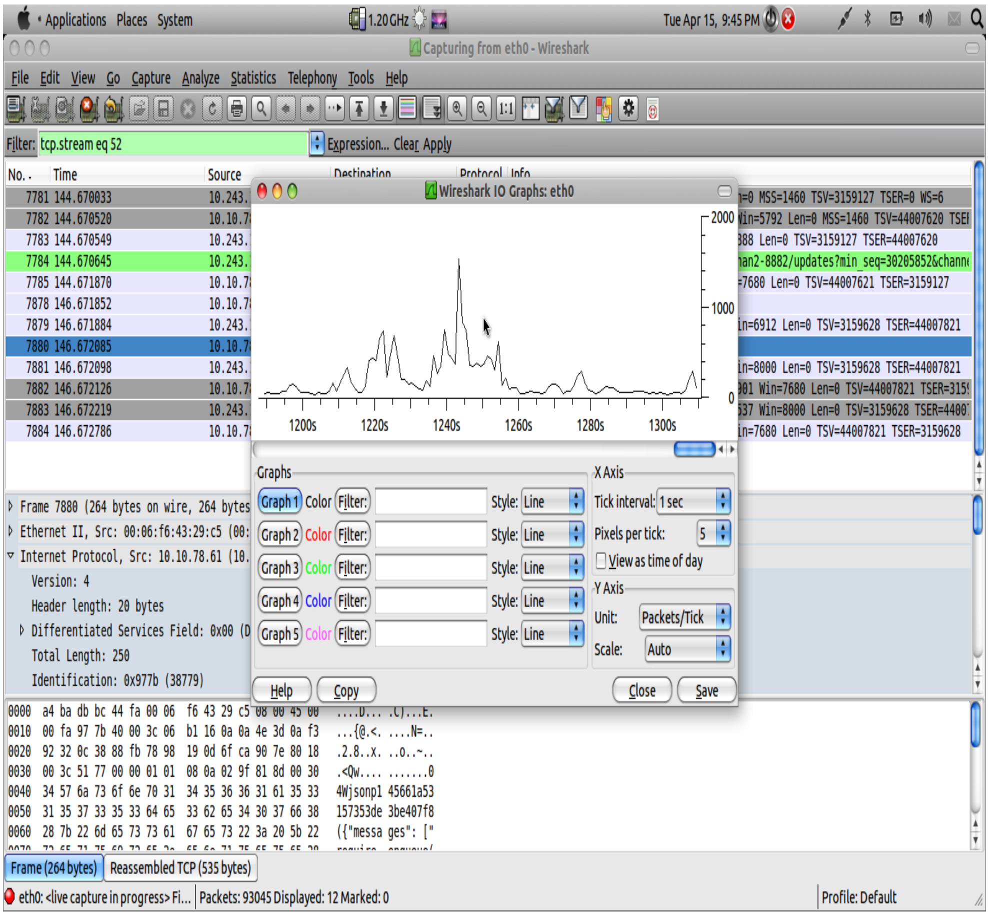
RESULTS AND CONCLUSIONS



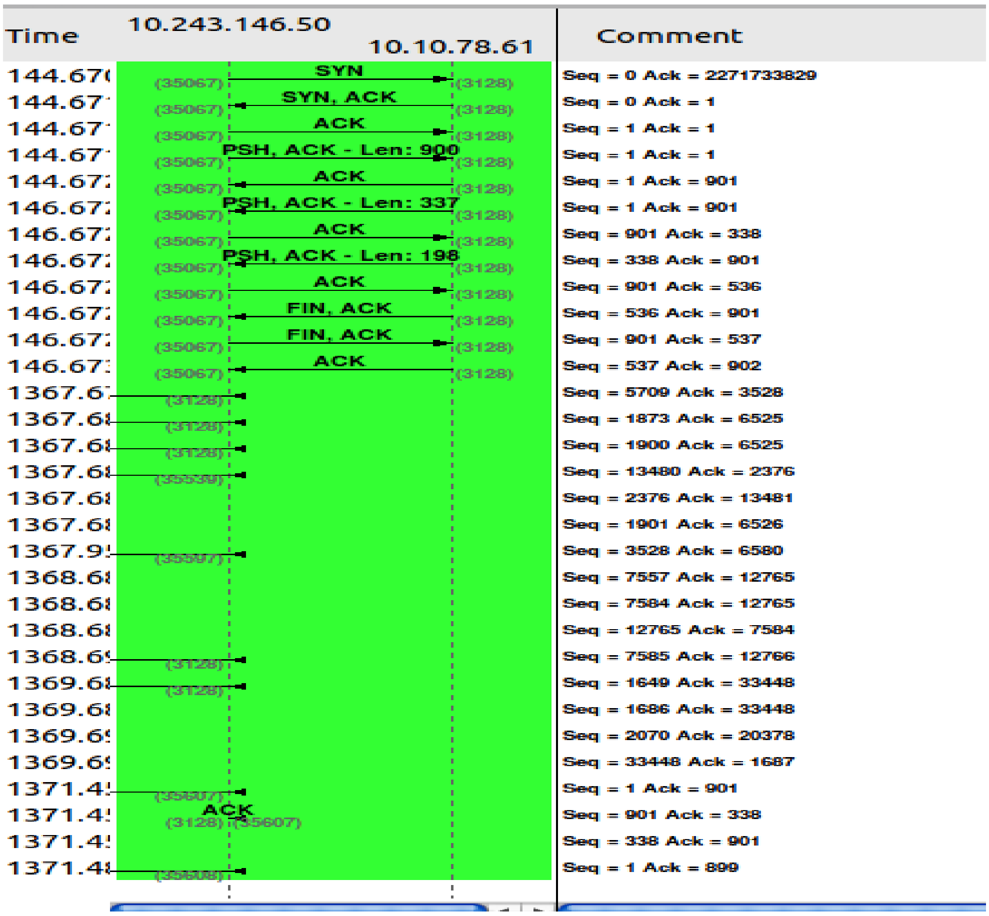
Capturing and analyzing network traffic using Wireshark



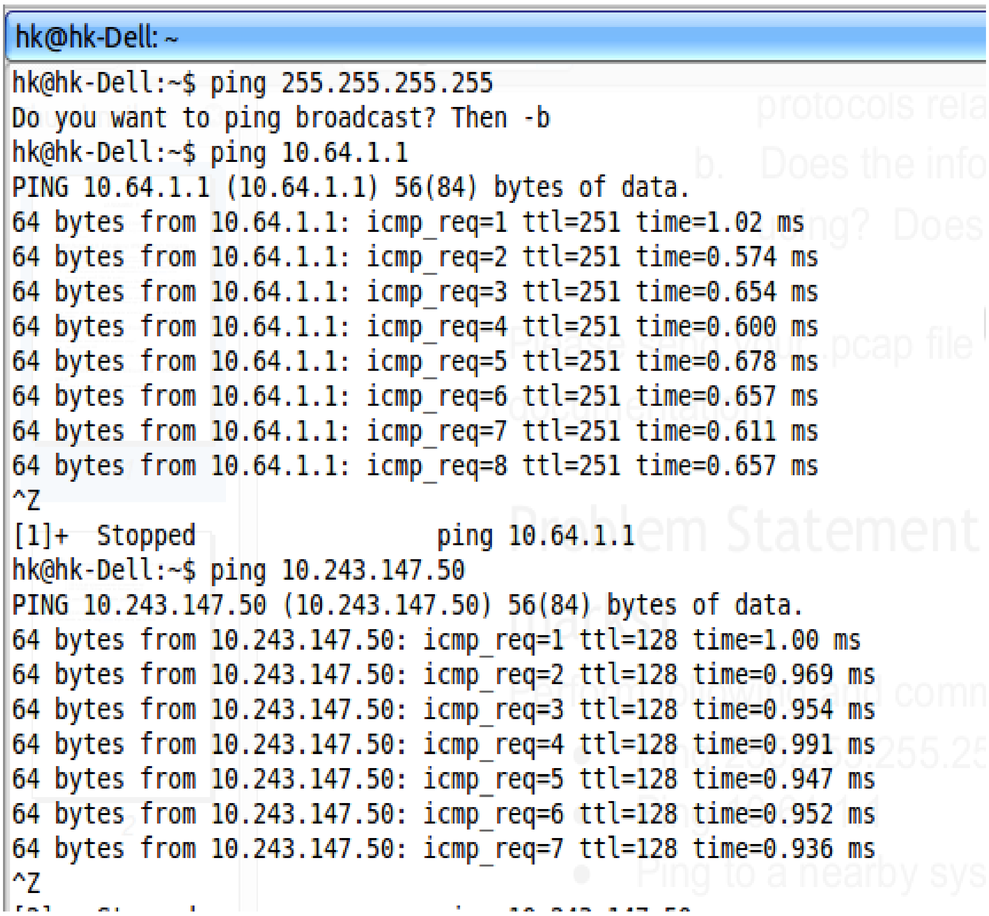
Analysis of various packet Headers using Wireshark



Analysis of IO using Wireshark (IO graph)



TCP packets transfer analysis



Ping 255.255.255.255 , 10.64.1.1 and a nearby system

traceroute to stanford.edu (171.67.215.200), 30 hops max, 60 byte packets

1	static.121.168.4.46.clients.your-server.de	46.4.168.121	de	**0.873** ms	**0.963** ms	**0.987** ms
2	hos-tr3.juniper2.rz13.hetzner.de	213.239.224.65	de	**11.924** ms		
	hos-tr4.juniper2.rz13.hetzner.de	213.239.224.97	de	**12.061** ms		
	hos-tr1.juniper1.rz13.hetzner.de	213.239.224.1	de	**0.113** ms		
3	core21.hetzner.de	213.239.245.81	de	**0.246** ms		
	core22.hetzner.de	213.239.245.121	de	**15.602** ms		
	core21.hetzner.de	213.239.245.81	de	**0.246** ms		
4	core11.hetzner.de	213.239.245.225	de	**2.747** ms		
	core11.hetzner.de	213.239.245.221	de	**4.133** ms		
	core12.hetzner.de	213.239.245.29	de	**2.840** ms		
5	juniper4.rz2.hetzner.de	213.239.203.138	de	**2.863** ms		
	juniper4.rz2.hetzner.de	213.239.245.26	de	**2.860** ms		
	juniper4.rz2.hetzner.de	213.239.203.138	de	**2.863** ms		
6	30gigabitethernet1-3.core1.ams1.he.net	195.69.145.150	nl	**13.952** ms	**13.796** ms	**13.777** ms
7	100ge9-1.core1.lon2.he.net	72.52.92.213	us	**22.869** ms	**22.269** ms	**22.252** ms
8	100ge1-1.core1.nyc4.he.net	72.52.92.166	us	**84.734** ms	**84.756** ms	**84.695** ms
9	100ge7-2.core1.chi1.he.net	184.105.223.161	us	**106.092** ms	**105.941** ms	**106.071** ms
10	10ge11-4.core1.pao1.he.net	184.105.222.173	us	**164.411** ms	**159.352** ms	**159.431** ms
11				*	*	*

Trace the route to http://www.stanford.edu/ using traceroute