Genie Snoop lab

Laboration in data communication
GenieLab
Department of Information Technology, Uppsala University

Overview

This lab deals with network layers, services and HTTP transactions as well as practicing (linux) network tools and command usage.

Administration

Student 1

Name: _____

Email: _____

Personal number: _____

Course UID: _____

Student 2

Name: _____

Email: _____

Personal number: _____

Course UID: _____

Agreement

I/we have independently worked on the following assignment solution. In the case of two people, we have both taken part in creating the solution, according to the assignment specification.

Sign 1: _____

Sign 2: _____

General

Course instance (e.g. Datakom DV1): _____

Date: _____

Note to the lab assistant

_____

_____

_____

_____

_____

_____

Comments from the lab assistant

_____

_____

_____

_____

_____

_____

Grade: _____

Sign: _____

1 About the lab

Objectives

- Understanding of network layers (e.g. application, transport, link)
- Understanding the role of services offered by each layer
- To gain deeper insight into web transactions using HTTP, and the resulting network traffic
- Practicing network tools and command usage

Reading instructions

In the reading instructions below, the following abbreviations are used to denote different books.

CN Computer Networking - a top-down approach featuring the Internet (3rd edition) - Kurose, Ross

DS Distributed Systems - concepts and design (4th edition) - Coulouris, Dollimore, and Kindberg

- HTTP

  CN 2.2

  DS Pages 13, 160-164

- Routing

  CN 4.1-4.2

  DS Sections 3.3.5 and 3.4.3

- IPv4

  CN 4.4, especially 4.4.1 (datagram format)

  DS Pages 89-96

- Link layer

  CN 5.1

  DS Pages 112-118

- MAC and ARP

  CN 5.4

  DS Section 3.4.2

Theoretical and practical questions

There are a number of questions marked with a (T), these are theoretical questions which do not require you to use your computer. I.e., these questions can be done after the lab in case you run out of time in the lab room.

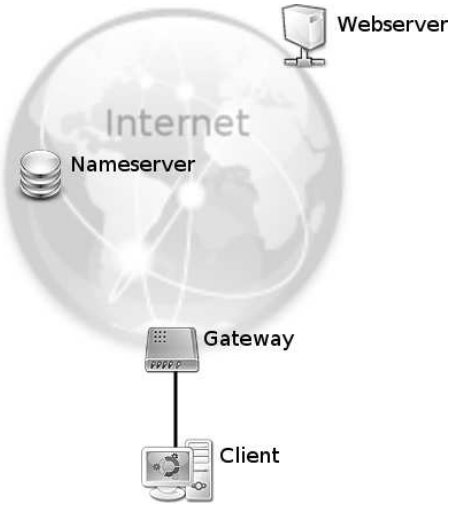

Figure 1: The network setup

2 Lab description

General

In this lab you will use a network monitoring program ("sniffer") to listen to all packets that are sent on the wire connected to a certain network card on your machine. It can be seen as a digital form of microphone that "records" everything that goes through the wire. Due to the network setup (switched network) you will only see packets to and from your computer + broadcast packets. That is, you can not easily eavesdrop your neighbours traffic in this kind of network by sniffing from your computer's network card.

The network, which in fact is a virtual network, will consist of a number of machines. The computer which is to be seen as the one you are sitting at (and the one which most of the work is done with) is named 'client', and is available through a terminal. Later on during the lab you will also access a machine named 'webserver' through another terminal-window. In addition we also have access to a gateway and a nameserver on the network.

The sniffer program, Ethereal, contains a graphical interface which allows the client to see the captured packets in a systematic and structured way. The packets

are seen in the order they were captured, and the program automatically parses and presents them in a tree-view based of the headers in the packet. This makes it easier for the client to quickly get an overview of the packets that are captured.

However, it is important to realize that the packets that are captured are really just chunks of bytes (as presented in the bottom window), and that Ethereal is just a tool for presenting the data.

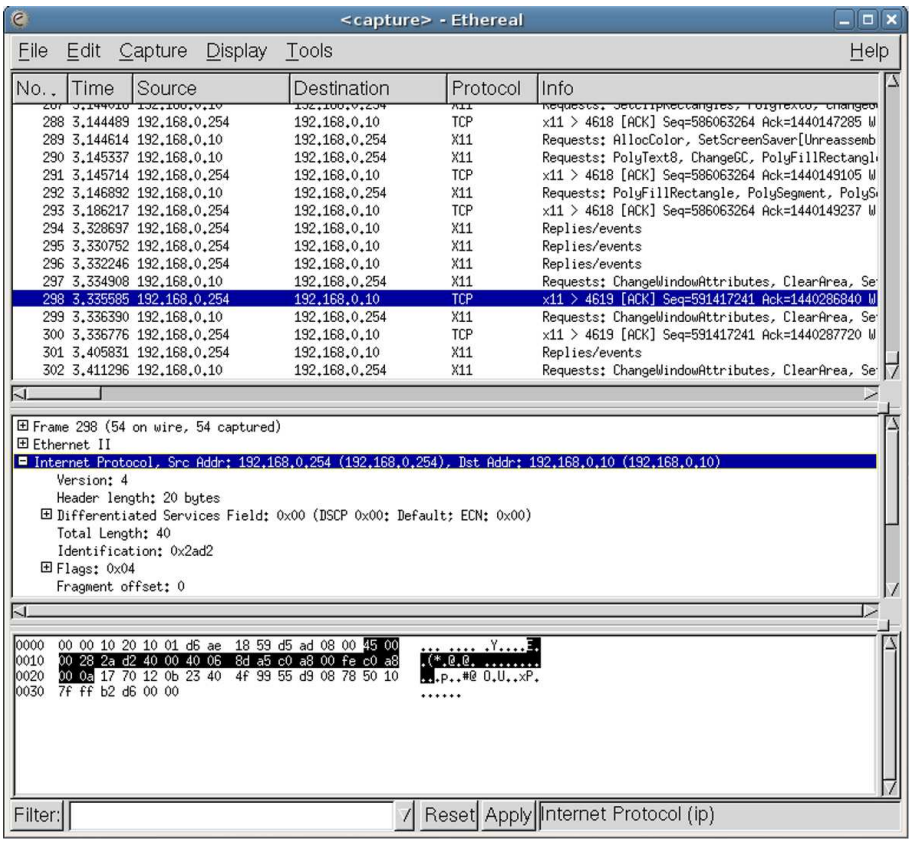

Figure 2: Ethereal screen dump

Startup

- Begin the lab by clicking the icon named “Start Snoop lab” once. Two terminal windows will appear on the screen, they are connected to the computers ‘client’ with text in green and ‘webserver’ with text in red, respectively.

- Login to the machine ‘client’. Enter the username **root** and password **lab**. This is the login and password used for all machines in this lab. *Do not log in to the machine ‘webserver’ yet.*

- Start a sniffer program in the 'client' terminal. This is done by entering the command `ethereal &` in the 'client' window.

Stage 1 - Packet examination

Introduction - How to capture packets

- In Ethereal: Click the menu-item Capture->Start, select interface eth0 and press the *OK* button. The program will now listen to all packets that are sent to/from the computer 'client'. *Make sure you select the interface eth0!*
- Go back to the 'client' terminal, start a webbrowser by entering `mozilla &` and go to http://www.genielab.net/ . You will see a new page appear in your web browser.
- **Wait approximately 20 seconds** to make sure you capture all packets before performing next step.
- Switch back to Ethereal, and press the *Stop* button to stop collecting data. Now the upper table should be filled with data. Each row of the table corresponds to a packet. To see more detailed information about each packet mark the row and look at the middle window. You can click the plus sign (+) to expand a row to see more detailed information. When marking a row in the middle window, you will see the corresponding portion of the HEX-dump highlighted in the bottom window.

  You might see some rows saying "Unknown WCCP message" or similar, this is due to a bug in Ethereal. Mark one of these packets, select the "Tools->Decode as" menu option and select DNS from the list.

Assignment

You now know how to capture packets with ethereal, it is time to put that knowledge to use! Visit the web page at http://www.genielab.net/lab1/
You will see a page with two links. Capture data with Ethereal while accessing page1. You should use the data from Ethereal to answer these questions:

1.1 (a) Which transport protocol is used for web traffic?

(b) In which IP header field can you get that information?

(c) Which value (number) identifies this transport protocol?

1.2 (a) Which application protocol is used for web traffic?

(b) When your computer receives a TCP/IP packet, how does it decide to which application it should send the packet? E.g. to choose whether your web server or mail server should handle the packet?

To answer question 1.3 it might help you to also capture the traffic for page2 at www.genielab.net/lab1/ and compare with the capturing of page1, you will have to start another instance of ethereal and capture the traffic for page2 in this window. Compare the data in the different ethereal-windows to find the answer.
When capturing the traffic for page2, make sure you enter the data in the field and press the submit button. Save one of these captures (click File->Save...), you will need it for Stage 3, Question 3.3.

1.3 When the web server receives a packet, how does it determine which operation to perform?

1.4 Why are there so many packets (other than those that contain HTTP data) sent to/from your computer when clicking on a link? Identify the other packets. Why is it important to wait 20 seconds before ending the capture, as you did since you carefully read the instructions?

1.5 (T) Draw a picture of a complete HTTP frame (including all headers, like the IP header, etc.) and mark where the different headers and the data are located. Do not draw individual fields in the headers. Draw the packet as a rectangular box.

1.6 Visit a page with pictures, more specifically, http://www.genielab.net/lab1/volvo.html. Determine the nature of the *HTTP session*, is it persistent or non-persistent? (Consult the textbook if you don't know what this means.) Do you see any contradiction regarding the nature of the session? Discuss this with your teaching

assistant if you don't discover it by yourself. Give an exhaustive motivation to what the contradiction may be. Motivate why it is like this; what are the benefits and drawbacks?

Note: To answer this question you must study the capture in great detail.

1.7 How much overhead does the entire transfer have for the page /lab1/overhead.html. Calculate the total size of all packets transferred (all packets that you captured), from Ethernet frame to data. The total packet size can be found in in the middle window of Ethereal, by expanding the Frame x (where x is a number). The payload is considered to be the *HTML code* that is sent to the client. Present your calculations and give an answer in percentage (%).

Stage 2 - Address resolution at different layers

You will be visiting the link http://www.genielab.net/search which is a minimalistic search site (like http://www.google.com/). Search for pages with cars using the search word "cars". You will get a hit list of three entries. Pick the one you like

most and visit that URL (Hint: copy/paste if the address is long). The URL will be shown in the field at top of the web browser.

Assignment

2.1 Where is the page located? Answer with the host name. (Hint: use information from the URL)

_____

2.2 Start a new capture in ethereal and use the DNS (Domain Name System) to resolve the IP address of the host. This is done with the `host` command.

_____

2.3 (a) Consult ethereal and determine which transport protocol is used for DNS.

_____

(b) Which number is this transport protocol represented by?

_____

(c) (T) Why do you think this protocol has been chosen?

_____

_____

_____

2.4 Capture the packets in ethereal while retrieving the web page of your choice (inside the genielab.net domain). Look at the "Destination Address" field in the Ethernet frame of a packet sent from the machine ‘client’. Try the `arp -n` command, to see the ARP (Address Resolution Protocol) cache.

(HWaddress == MAC address)

(a) Which IP address does the MAC address resolve to? (Ignore the 10.20.10.250 address.)

(b) Look at the IP header of the packet, especially the field "Target IP". Why do the destination address in the Ethernet II frame (which you resolved to an IP address in (a)) and the IP header differ?

If the ARP cache is empty do the request again, it expires after a couple of minutes.

(c) (T) Why do the ARP entries expire?

(d) (T) Why are they cached at all?

2.5 From question 2.4 you will see that the packet is not sent directly to the destination. You can confirm this by running `traceroute www.genielab.net`, which prints the route to the host.

(a) How does the sender decide whether to send a packet directly or not? (Hint: `route` command)

(b) (T) Why is routing/IP forwarding an important concept? Explain and give an example where routing/IP forwarding is necessary.

Stage 3 - The big picture

In this step you will capture packets at the webserver. The webserver is located on 10.20.0.1.

Important! Before accessing a web page, delete the ARP entry (if any) on *'client'* for your default gateway (10.20.2.30), `arp -d 10.20.2.30` in the green 'client' window.

Login to the webserver (red window) with the clientname **root** and password **lab**. Then start a sniffer for capturing data by issuing the command `ethereal &` in the red 'webserver" window. Capture data on the interface eth0.

Assignment

3.1 Capture data at both ‘webserver’ and ‘client’, while accessing a web page. Compare the data from both computers. For simplicity, choose one packet (e.g. the first HTTP packet) and compare the same packet on both computers. How do they differ?

3.2 Compare the packets that are captured now with the packets you saved in Stage 1. To load a previously saved capture click File->Load, it might help to open a new ethereal window so you can see both captures at the same time. Remember to load the saved capture from the same terminal as you saved it, i.e. ‘client’. Are there any new kinds of packets appearing? What are their purposes?

If you cannot see any difference, read the introduction in stage 3 and conduct those steps again before trying to repeat the capture.

3 The report

You should answer all of the questions above. Also make sure that you covered all subquestions.

You may write in English or in Swedish.

Hand in this printed out lab with your answers filled (in a legible style) to the lab assistant's pigeonhole. This should be done before the date announced at the lab page for your course instance.