VIT
UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)
Vellore - 632 014, Tamil Nadu, India.

School of Information Technology and Engineering

Fall-2015

B.Tech (Information Technology) - V Semester

ITE 304 Computer Networks Lab

Cycle sheet-3

UDP PROGRAMS

1. Implement echo server and client in java using UDP sockets.

2. Write a program to implement a text based message transfer from client to server process using UDP.

3. Implement a chat server and client in java using UDP sockets.

4. Implement a DNS server and client in java using UDP sockets.

5. Find the logical address of a host when its physical address is known (RARP protocol) using UDP.

6. Find the physical address of a host when its logical address is known (ARP protocol) using UDP.

7. Implement Client - Server communication to access Date using UDP in Java.

WIRESHARK EXERCISES

A) Wireshark Lab: HTTP

1. The Basic HTTP GET/response interaction. Let's begin our exploration of HTTP by downloading a very simple HTML file - one that is very short, and contains no embedded objects.

Do the following:

1. Start up your web browser.

2. Start up the Wireshark packet sniffer, as described in the Introductory lab (but don’t yet begin packet capture). Enter “http” (just the letters, not the quotation marks) in the display-filter-specification window, so that only captured HTTP messages will be displayed later in the packet-listing window. (We’re only interested in the HTTP protocol here, and don’t want to see the clutter of all captured packets).

3. Wait a bit more than one minute (we’ll see why shortly), and then begin Wireshark packet capture.

4. Enter the following to your browser

i. http://intranet.vit.ac.in/

5. Your browser should display the very simple, one-line HTML file.

6. Stop Wireshark packet capture.

By looking at the information in the HTTP GET and response messages, answer the following questions. When answering the following questions, you should print out the GET and response messages (see the introductory Wireshark lab for an explanation of how to do this) and indicate where in the message you’ve found the information that answers the following questions.

1. Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running?

2. What languages (if any) does your browser indicate that it can accept to the server?

3. What is the IP address of your computer? Of the intranet.vit.ac.in server?

4. What is the status code returned from the server to your browser?

5. When was the HTML file that you are retrieving last modified at the server?

6. How many bytes of content are being returned to your browser?

7. By inspecting the raw data in the packet content window, do you see any headers within the data that are not displayed in the packet-listing window? If so, name one.

B) Wireshark Lab: ICMP

1. ICMP and Ping

Let's begin our ICMP adventure by capturing the packets generated by the Ping program. The Ping program is simple tool that allows anyone (for example, a network administrator) to verify if a host is live or not. The Ping program in the source host sends a packet to the target IP address; if the target is live, the Ping program in the target host responds by sending a packet back to the source host. As you might have guessed (given that this lab is about ICMP), both of these Ping packets are ICMP packets.

Do the following :

- Start up the Wireshark packet sniffer, and begin Wireshark packet capture.
- Use The ping command to ping –n 10 http://intranet.vit.ac.in/
- When the Ping program terminates, stop the packet capture in Wireshark.

1. What is the IP address of your host? What is the IP address of the destination host?

2. Why is it that an ICMP packet does not have source and destination port numbers?

3. Examine one of the ping request packets sent by your host. What are the

ICMP type and code numbers? What other fields does this ICMP packet have? How many bytes are the checksum, sequence number and identifier fields?

4. Examine the corresponding ping reply packet. What are the ICMP type and code numbers? What other fields does this ICMP packet have? How many bytes are the checksum, sequence number and identifier fields?

C) Wireshark Lab: Ethernet and ARP

1. Capturing and analysing Ethernet frames.

Do the following:

1. First, make sure your browser's cache is empty.

2. Start up the Wireshark packet sniffer

3. Enter the following URL into your browser
   http://intranet.vit.ac.in/

4. Stop Wireshark packet capture. First, find the packet numbers (the leftmost column in the upper Wireshark window) of the HTTP GET message that was sent from your computer to http://intranet.vit.ac.in as well as the beginning of the HTTP response message sent to your computer by gaia.cs.umass.edu.

Answer the following questions:

1. What is the 48-bit Ethernet address of your computer?

2. What is the 48-bit destination address in the Ethernet frame? Is this the

   Ethernet address of intranet.vit.ac.in? What device has this as its Ethernet address?

3. What is the hexadecimal value of the CRC field in this Ethernet frame?

D) Wireshark Lab: TCP

1. Capturing a bulk TCP transfer from your computer to a remote server.

Do the following:

Start up your web browser. Go the http://www.vit.ac.in

Click the link attendance.

Next go to http://www.vit.ac.in/admissions/Programmes.asp

Stop Wireshark packet capture.

A first look at the captured trace

Before analyzing the behavior of the TCP connection in detail, let's take a high level view of the trace.

First, filter the packets displayed in the Wireshark window by entering "tcp" (lowercase, no quotes, and don't forget to press return after entering!) into the display filter specification window towards the top of the Wireshark window.

Answer the following questions:-

1. What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu? To answer this question, it's probably easiest to select an HTTP message and explore the details of the TCP packet used to carry this HTTP message, using the "details of the selected packet header window.

2. What is the IP address of www.vit.ac.in? On what port number is it sending and receiving TCP segments for this connection?

If you have been able to create your own trace, answer the following question:

3. What is the IP address and TCP port number used by your client computer (source) to www.vit.ac.in?

Since this lab is about TCP rather than HTTP, let's change Wireshark's "listing of captured packets" window so that it shows information about the TCP segments containing the HTTP messages, rather than about the HTTP messages. To have Wireshark do this, select Analyze- >Enabled Protocols. Then uncheck the HTTP box and select OK.

Answer the following questions for the TCP segments:

4. What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and www.vit.ac.in? What is it in the segment that identifies the segment as a SYN segment?

5. What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN? What is the value of the ACKnowledgement field in the SYNACK segment? How did gaia.cs.umass.edu determine that value? What is it in the segment that identifies the segment as a SYNACK segment?