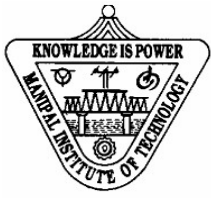
MANIPAL INSTITUTE OF TECHNOLOGY

Manipal – 576 104

DEPARTMENT OF COMPUTER SCIENCE & ENGG.



CERTIFICATE

This is to certify that Ms./Mr. ……...…… Reg. No. …..…… Section: …… Roll No: ……... has satisfactorily completed the lab exercises prescribed for CSE 3113: Computer Networks Lab[CSE3113] of Third Year B. Tech. Degree in Computer Science & Engg. at MIT, Manipal, in the academic year 2016-2017..

Date: …….....

Signature
Faculty in Charge

Signature
Head of the Department

CONTENTS

LAB NO	TITLE	PAGE NO	REMARKS
	Course Objectives and Outcomes	i	
	Evaluation Plan	i	
	Instructions to the Students	ii	
1	Basics of Linux Programming Environment	1-10	
2	Socket Programming using UDP Protocol	11-18	
3	Socket Programming using TCP Protocol	19-25	
4	Implementation of TCP/USP Concurrent Servers	26-31	
5	Programs on Synchronous I/O Multiplexing using select() system call	32-37	
6	Mini Project	38	
7	Study of Application Layer Protocols	39-47	
8	Study of Transport Layer Protocols & Network Layer Protocols	48-59	
9	Data Link Layer Protocols	60-67	
10	Hands on Experiments-1: LAN & Network Services	68-70	
11	Hands on Experiments-2: Inter-networking & Network Services	71-73	
12	Mini Project Evaluation	74	
13	References	75	

Course Objectives

- To Understand Network Programming using Linux System Calls.
- To Develop skills in Network Monitoring and Analysis using various tools.
- To Understand development of Network Applications.
- To Design and Deploy Computer Networks.

Course Outcomes

At the end of this course, students will be able to

- Learn to use Network Related commands and configuration files in Linux Operating System.
- Learn to Develop Network Application Programs.
- Analyze Network Traffic using network Monitoring Tools

Evaluation plan

- Internal Assessment Marks : 60 Marks

✓ Continuous evaluation component :5 Evaluations *8M = 40 marks

✓ Mini project : = 20 marks

✓ The assessment will depend on punctuality, program execution, maintaining the observation note and answering the questions in viva-voce

- End semester assessment : 40 Marks

✓ Duration: 2 hours

✓ Total marks : Write up: 15 Marks

Execution: 25 Marks

INSTRUCTIONS TO THE STUDENTS

Pre- Lab Session Instructions

1. Students should carry the Class notes, Lab Manual and the required stationery to every lab session
2. Be in time and follow the Instructions from Lab Instructors
3. Must Sign in the log register provided
4. Make sure to occupy the allotted seat and answer the attendance
5. Adhere to the rules and maintain the decorum

In- Lab Session Instructions

- Follow the instructions on the allotted exercises given in Lab Manual
- Show the program and results to the instructors on completion of experiments
- On receiving approval from the instructor, copy the program and results in the Lab record
- Prescribed textbooks and class notes can be kept ready for reference if required

General Instructions for the exercises in Lab

- The programs should meet the following criteria:
    - Programs should be interactive with appropriate prompt messages, error messages if any, and descriptive messages for outputs.
    - Use meaningful names for variables and procedures.
- Copying from others is strictly prohibited and would invite severe penalty during evaluation.
- The exercises for each week are divided under three sets:
    - Lab exercises - to be completed during lab hours
    - Additional Exercises - to be completed outside the lab or in the lab to enhance the skill
- In case a student misses a lab class, he/ she must ensure that the experiment is completed at students end or in a repetition class (if available) with the permission of the faculty concerned but credit will be given only to one day's experiment(s).
- Questions for lab tests and examination are not necessarily limited to the questions in the manual, but may involve some variations and / or combinations of the questions.

THE STUDENTS SHOULD NOT...

- Bring mobile phones or any other electronic gadgets to the lab.
- Go out of the lab without permission.

LAB NO: 1 **Date:**

BASICS OF LINUX PROGRAMMING ENVIRONMENT

Objectives:

- Getting familiar with Linux Programming Environment.
- Understanding of network diagnostic tools like tcpdump and wireshark**.**

Prerequisites:

- Knowledge of the essential Linux commands.
- Knowledge of the usage of Linux system calls in C Programs.

Play Time with Linux:

- **Learn about following essential Linux commands:**

See the on-line manual pages for additional information on all commands.

man,passwd,pwd,ls,more,mv,rm,mkdir,rmdir,cd,cp,chmod,who,ps,kill,ctrl+c,cmp,grep,**cat,**

Redirection operators(> <)

Most of the above commands accept input from the system's *standard input* device (e.g., the keyboard) and send an output to the system's *standard output* device (e.g., the screen). Sometimes it is convenient to direct the output to another process as input for further processing, or to a file for storage. The *redirect* operator ">" directs the output to a file, as:

Eg.:

ls > *filename*

Pipe operator (|)

With the pipe operator "|", two commands can be concatenated as: **command1 | command2**, where the output of **command1** is redirected as the input of **command2**

(i) Knowledge on Usage of Linux System calls

Open (): opens a file in specified mode	Close (): closes specified file descriptor
Read (): read from a file	Write (): write to a file
Creat (): create a file	Fork (): create a child process
getpid, getppid - get process identification	getgid, getegid - get group identity
wait, waitpid, waitid - wait for process to change state	lseek - reposition read/write file offset

(Refer to Linux **man pages** for more information)

- ## Network daemons and services

A daemon is a process running in the background of the system. Many TCP/IP services (e.g., Telnet) are handled by a daemon called **inetd**. Rather than running several network-related daemons, the **inetd** daemon works as a dispatcher and starts the necessary server processes when requests arrive. When a client wants a particular service from a remote server, the client contacts the **inetd** daemon through the server's well- known port number, which prompts **inetd** to start the corresponding server process.

The network daemons managed by **inetd** are specified in a configuration file called **/etc/inetd.conf**. Each service has a line in the file defining the network daemon that provides the service and its configuration parameters. One can comment a line, i.e., insert# at the beginning of the line, to disable the corresponding service. Note that there are some stand-alone network daemons that are not managed by **inetd**. For example, web service is provided by the **httpd** daemon, and DNS service is provided by the **named** daemon.

In Latest Linux distributions, **xinetd** replaces inetd, adding stronger security and more functionality. **Xinetd** uses a simple common configuration file **/etc/xinetd.conf**. In addition, each service managed by **xinetd** uses an individual configuration file in the **/etc/xinetd.d** directory.

Well-known port numbers are defined in the **/etc/services** file. A server can handle multiple clients for a service at the same time through the same well-known port number, while a client uses an ephemeral port number. The uniqueness of a communication session between two hosts is preserved by means of the port number and IP address pairs of the server and client hosts.

- ## Network configurations files

When a host is configured to boot locally, certain TCP/IP configuration parameters are stored in appropriate local disk files. When the system boots up, these parameters are read from the files and used to configure the daemons and the network interfaces. A parameter may be changed by editing the corresponding configuration file.

In addition to */etc/services* and */etc/inetd.conf* discussed above, we now list other network configuration files.

Linux Configuration Files	Purpose
/etc/hosts	Stores the host name of this machine and other machines.
/etc/sysconfig/network	Stores the host name and the default gateway IP address
/etc/sysconfig/network- scripts/ifcfg-	Stores the IP address of the first
eth0	Ethernet interface
/etc/default-route	Stores a default gateway, i.e., the IP address or the domain name of the default router.
/etc/resolv.conf	Stores the IP addresses of the DNS servers
/etc/nsswitch.conf	Configures the means by which host names are resolved.

- **Understanding Basic Networking Commands in Linux**

Configuring a network interface

The **netstat** command can be used to display the configuration information and statistics on a network interface. The same command is also used to display the host routing table. Several **netstat** options are listed below that will be used frequently in the experiments.

netstat -*a***:** Shows the state of all sockets, routing table entries, and interfaces.
netstat -*r***:** Displays the routing table.
netstat -*i* : Displays the interface information.
netstat -*n***:** Displays numbers instead ofnames.
netstat -*s***:** Dis plays per-protocol statistics.

The **ifconfig** command is used to configure a network interface. The following options are used for the reconfiguration of the IP address and network mask.

ifconfig -*a :* Shows the states of all interfaces in the system.

ifconfig <*interface name*> *down* : Disables the network interface, where interface name is the name of the Ethernet interface.

ifconfig <*interface name*> <*new IP address*> *up* : Assigns a new IP address to the interface and brings it up.

ifconfig <*interface name* > *netmask* <*new netmask*> : Assigns a new network mask for the interface.

- **Knowledge of Network Diagnostic tools**

Diagnostic tools are used to identify problems in the network, as well as to help understand the behavior of network protocols. We will use the tcpdump and Wireshark tools extensively in the experiments. Students need to learn usage of these tools before starting with experiments.

Tcpdump

Tcpdump is a network traffic sniffer built on the packet capture library *libpcap*. While started, it captures and displays packets on the LAN segment. By analyzing the traffic flows and the packet header fields, a great deal of information can be gained about the behavior of the protocols and their operation within the network. Problems in the network can also be identified. A packet filter can be defined in the command line with different options to obtain a desired output.

One of the many good on-line resources is available about tcpdump is at:
https://danielmiessler.com/study/tcpdump/

Wireshark

Wireshark is a network protocol analyzer built on the packet capture library *pcap*. In addition capturing network packets as in *Tcpdump*, Wireshark provides a user friendly graphical interface, and supports additional application layer protocols. Wireshark can also import pre-captured data files from other network monitoring tools, such as *Tcpdump* and *Sniffer*.
Refer to https://www.wireshark.org/docs/ for more information.

Lab Exercises:

Q1.1) Login to the system. Enter your user account with login ID and the password. Get acquainted with the Desktop environment, the Linux commands, text editors, and the man pages.

Q1.2)

After logging in, open a command window, Show your login ID by typing **whoami** in the console.

Create a directory of your own, using **mkdir** *name_of your_directory*. Change to your directory, using **cd** *name_of_your_directory*. You can save your data files for all your laboratory experiments here.

Open another command window. Run **pwd** in this and the previously opened consoles. Save the outputs in both consoles.

What is the default directory when you open a new command window? What is your working directory?

Q1.3)

Run **ps -e** to list the processes running in your host. After starting a new process by running telnet in another command window, execute **ps -e** again in a third window to see if there is any change in its output. Find the process id of the telnet process you started, by: **ps -e | grep telnet.** Then use **kill** process id of telnet to terminate the **telnet** process.

Q1.4)

Is the Internet service daemon, **xinetd**, started in your system? Is **inetd** started in your system? Why? Document issued commands and its output.

Q1.5)

Display the file */etc/services* on your screen, using: **more /etc/services.** Then in another console, use the redirect operator to redirect the **more** output to a file using,

more /etc/services > ser_more

Compare the file ***ser_more*** with the original *more* output in the other command window. Copy */etc/services* file to a local file named *ser_cp* in your working directory, using **cp /etc/services ser_cp**. Compare files ser_more and ser_cp, using

cmp ser_more ser_cp. Are these two files identical?

Concatenate these two files using **cat ser_more ser cp > ser_cat**.

Display the file sizes using **ls -l ser***. Save the output. What are the sizes of files *ser_more, ser_cp, and ser_cat?*

Q1.6)

Write the **ls** output you saved in this exercise and answer the above questions.

Q1.7)

Read the **man** pages for the following programs and learn the usage

Arp	**Arping**	**Ifconfig**	**Tcpdump**
dig	**Netstat**	**Route**	**Wireshark**
Ping			

Explain each of the above commands briefly. *Two or three sentences* per command would be adequate. Document the output observed while working with above commands.

Additional Exercises:

Q1.8)

Capture Network Packets using tcpdump and classify it based on TCP and UDP protocols.

Q1.9)

Capture Network Packets using Wireshark and classify it based on various protocols.

[OBSERVATION SPACE – LAB1]

[OBSERVATION SPACE – LAB1]

[OBSERVATION SPACE – LAB1]

[OBSERVATION SPACE – LAB1]

[OBSERVATION SPACE – LAB1]

LAB NO: 2 **Date :**

Socket Programming using UDP Protocol

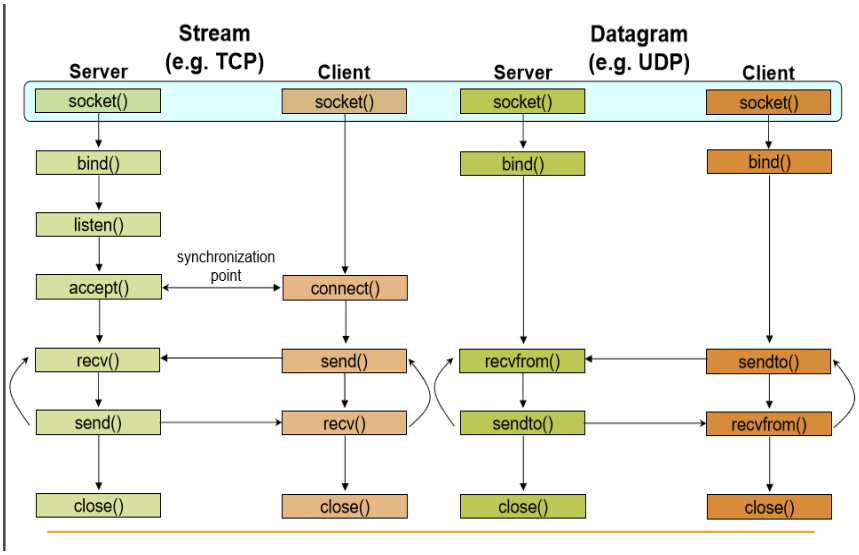
Objectives:

- Understand UDP Client-Server Socket-Programming
- Learn to edit the source code using ***Vi Editor***, Compilation and debugging in Linux Environment.

Prerequisites:

- Basic Understanding of Connection Less Services of UDP.

Note : Use following General Outline in coding TCP/UDP Client Server Coding



Lab exercises:

Q2.1)

Write an Echo Client -Server program using UDP (Refer to Given in Text.1 Table 2.22 and Table 2.23 at Page no. 118-121). Initially, you will run both client and server on the same machine. Once the programs are running correctly, you can pair with one of the other teams and test that your client runs with their server and vice-versa. In this case make necessary changes in the codes appropriately**.**

Q2.2)

Write a UDP Math server that handles multiple Clients. Server computes mathematical operations specified by clients. Client sends operands and operator to the server; server returns the result to clients for display. Also display server process ID (PID) in each transaction.

Additional Exercises:

Q2.3)

Write a Program using UDP Socket to implement a simple UDP peer to peer chat between two processes**.**

Q2.4)

Using Wireshark capture packets exchanged in the above Q2.3 Program and analyze them.

[OBSERVATION SPACE – LAB2]

[OBSERVATION SPACE – LAB2]

[OBSERVATION SPACE – LAB2]

[OBSERVATION SPACE – LAB2]

[OBSERVATION SPACE – LAB2]

[OBSERVATION SPACE – LAB2]

[OBSERVATION SPACE – LAB2]

LAB NO: 3 **Date :**

Socket Programming using TCP Protocol

Objectives:

- Understand TCP Client-Server Socket-Programming

Prerequisites:

- Basic Understanding of Connection Oriented Services of TCP.

Lab exercises:

Q3.1)

Write an Echo Client- Server program using TCP (Given in Text.1 Table 2.24 and Table 2.25 at Page no. 125-128) Initially, you will run both client and server on the same machine. Once the programs are running correctly, you can pair with one of the other teams and test that your client runs with their server and vice -versa. In this case make necessary changes in the codes appropriately.

Q3.2)

Write a TCP Server program that handles multiple Clients. Server computes square of a number sent by clients and returns the result to clients for display. Also display server process ID (PID) in each transaction.

Q3.3)

Write a Client/Server based application using TCP to execute the program at remote server. [The client sends the executable file to the server, server executes the file , stores the result in a file and sends back to the client. Client displays the result.]

Additional Exercises:

Q3.4)

Using Wireshark observe Three Way Handshaking Connection Establishment, Data Transfer and Three Way Handshaking Connection Termination in above Q3.3 program.

[OBSERVATION SPACE- LAB 3]

[OBSERVATION SPACE – LAB 3]

[OBSERVATION SPACE – LAB 3]

[OBSERVATION SPACE – LAB 3]

[OBSERVATION SPACE – LAB 3]

[OBSERVATION SPACE – LAB 3]

[OBSERVATION SPACE – LAB 3]

LAB NO: 4 **Date :**

IMPLEMENTATION OF TCP/UDP CONCURRENT SERVER

Objectives:

- To understand how Concurrent server handles multiple clients requests at a time.

Prerequisites:

- Understanding of Multiprocessing/Multitasking Concepts of Operating System.

Lab Exercises:

Q4.1)

Develop a Concurrent Time Server application using UDP to execute the program at remote server., [Client sends a time request to the server, server sends its system time back to the client. Client displays the result. Hint: use *fork()* at server]

Q4.2)

Develop concurrent file server which servers file requested by client if exists. If not then server sends appropriate message to the client.

Additional Exercises:

Q4.3)

Modify the above Q4.2 program where server sends its process ID (PID) to clients for display along with requested file.

[OBSERVATION SPACE-LAB 4]

[OBSERVATION SPACE – LAB 4]

[OBSERVATION SPACE – LAB 4]

[OBSERVATION SPACE – LAB 4]

[OBSERVATION SPACE – LAB 4]

[OBSERVATION SPACE – LAB 4]

LAB NO: 5 **Date :**

Programs on Synchronous I/O multiplexing using SELECT () system call

Objectives:

- **Understanding Synchronous I/O Multiplexing**

Lab exercises:

Q5.1)

Write a program to illustrate the use of 'select()' system call

Q5.2)

Develop a time server which shares system time to any clients in the network.

Additional exercises:

Q5.3)

Develop a multi user chat server using synchronous I/O multiplexing.

Q5.4)

Write a c program to implement packet capturing and filtering using raw sockets.

[OBSERVATION SPACE – LAB 5]

[OBSERVATION SPACE – LAB 5]

[OBSERVATION SPACE – LAB 5]

[OBSERVATION SPACE – LAB 5]

[OBSERVATION SPACE – LAB 5]

[OBSERVATION SPACE – LAB 5]

LAB NO: 6 **Date :**
MINI PROJECT

Objectives:

- Identify a Mini Project topic based on the Client-Server Concepts studied in this Lab
- Prepare a synopsis based on your Design.

Note:

- Mini Project may be implemented in groups of 3- 4 students
- Topics Suggested(But Not Limited to this list):

FTP Server, File Download Manager, LAN Chat Server, File Sharing Utility for LAN, Network Monitor for LAN etc.

- ***SUBMIT YOUR SYNOPSIS BEFORE STARTING OF WEEK 8 LAB.***

LAB NO: 7 **Date :**

STUDY OF APPLICATION LAYER PROTOCOLS

Objectives:

- To understand application layer functionality and protocols
- Know the meaning of process-to-process communication

Prerequisites:

- Brief idea about Application laye in Network Software Architecture and its protocols
- Working Knowledge of Wireshark, tcpdump, traceroute, ping toolsto capture and investigate some application layer protocols like HTTP, FTP, TELNET,SSH, SMTP, POP3, and DNS.

Lab exercises :

Q7.1)

Retrieve web pages using HTTP. Use Wireshark to capture packets for analysis. Learn about most common HTTP messages . Also capture response messages and analyze them. During the lab session, also examine and analyze some HTTP headers.

Q7.2)

Use FTP to transfer some files, Use Wireshark to capture some packets. Show that FTP uses two separate connections: a control connection and a data-transfer connection. The data connection is opened and closed for each file transfer activity. Also show that FTP is an insecure file transfer protocol because the transaction is done in plaintext.

Q7.3)

Use Wireshark to capture packets exchanged by the TELNET protocol. As in FTP, observe commands and responses during the session in the captured packets. Also show that TELNET is vulnerable to hacking because it send all data including the password in plaintext.

Q7.4)

Use Wireshark to capture packets exchanged by the SSH protocol. As in TELNET, observe commands and responses during the session in the captured packets. Also show that SSH is not vulnerable to hacking because it sends all data encrypted including the password .

Q7.5)

Investigate SMTP protocol in action. Sent an e-mail and, using Wireshark investigate the contents and format of the SMTP packet exchanged between client and the server. Explain the three phases in this SMTP session.

Additional Exercise

Q7.6)

Investigate the state and behavior of the POP3 protocol. Retrieve the mails stored in your mailbox at the POP3 server can observe and analyze the states of the POP3 and the type and the contents of the messages exchanged, by analyzing the packets through Wireshark.

Q7.7)

Analyze the behavior of the DNS protocol. In addition to Wireshark, several network utilities are available for finding some information stored in the DNS servers. Use dig utilities (which has replaced *nslookup*). Set Wireshark to capture the packets sent by this utility.

[OBSERVATION SPACE – LAB 7]

[OBSERVATION SPACE – LAB 7]

[OBSERVATION SPACE – LAB 7]

[OBSERVATION SPACE – LAB 7]

[OBSERVATION SPACE – LAB 7]

[OBSERVATION SPACE – LAB 7]

[OBSERVATION SPACE – LAB 7]

[OBSERVATION SPACE – LAB 7]

LAB NO: 8 **Date :**

STUDY OF TRANSPORT LAYER PROTOCOLS & NETWORK LAYER PROTOCOLS

Objectives

- Understand the working of transport layer and network layer protocols
- Understand end-to-end connection.

Prerequisites

- Knowledge of Transport Layer and Network layer protocol

Guidance:

- **UDP Header and Pseudoheader**

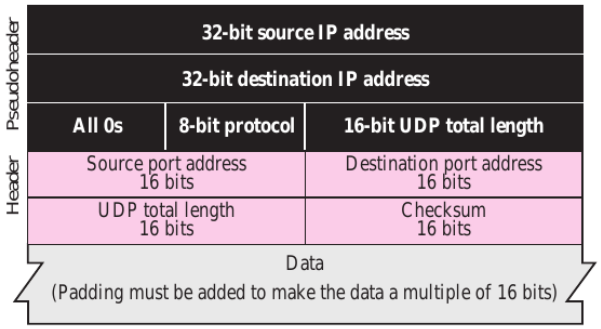


Fig 8.1 UDP Header and Pseudoheader

- **TCP Segment Format**

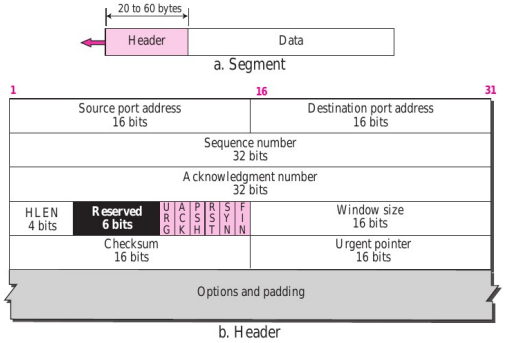


Fig 8.2 TCP Segment Format

- **General Format of ICMP Messages**

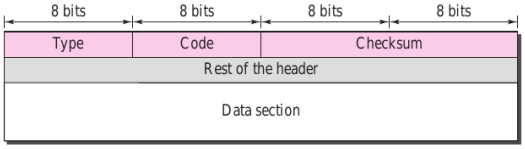


Fig. 8.3 General Format of ICMP Messages

- **ICMP Message Types**

Type	Code	Description	Query	Error
0	0	Echo reply	•	
3		Destination unreachable:		
	0	Network unreachable		•
	1	Host unreachable		•
	2	Protocol unreachable		•
	3	Port unreachable		•
	4	Fragmentation needed		•
	5	Source route failed		•
	6	Destination network unknown		•
	7	Destination host unknown		•
	8	Source host isolated		•
	9	Destination net prohibited		•
	10	Destination host prohibited		•
	11	Network unreachable for TOS		•
	12	Host unreachable for TOS		•
	13	Communication prohibited		•
	14	Host precedence violation		•
	15	Precedence cutoff in effect		•
4	0	Source quench		•

Type	Code	Description	Query	Error
5		Redirect		
	0	Redirect for network		•
	1	Redirect for host		•
	2	Redirect for TOS and Net		•
	3	Redirect for TOS and Host		•
8	0	Echo request	•	
9	0	Router advertisement	•	
10	0	Router solicitation	•	
11		Time exceeded		
	0	TTL equals 0 during transit		•
	1	TTL equals 0 during reassembly		•
12		Parameter problem		
	0	IP header bad		•
	1	Required option missing		•
13	0	Timestamp request	•	
14	0	Timestamp reply	•	
15	0	Information request	•	
16	0	Information reply	•	
17	0	Address mask request	•	
18	0	Address mask reply	•	

Fig. 8.4 ICMP Message Types

- **IP Datagram**

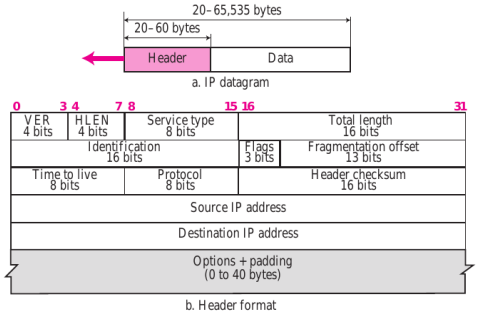


Fig. 8.5 IP Datagram

Lab exercises

Q8.1)

Use the Wireshark to capture UDP packets in action. Check the value of each field and check to see if the checksum has been calculated for the UDP.

Q8.2)

Use the Wireshark to study many features of TCP in detail. These features include reliability, congestion control, and flow control. Wireshark lets us see how TCP uses sequence and acknowledgement numbers in segments to achieve reliable data transfer. Also observe TCPs congestion algorithms (slow start, congestion avoidance, and fast recovery) in action. Another feature of TCP is flow control. See how the flow control in the direction of the receiver to sender in TCP is achieved using the value of *cwnd* advertised by the receiver.

Q8.3)

Capture and study ICMP packets generated by other utility programs such as *ping* and *traceroute*

- Type 0, code 0
- Type 3, code 3
- Type 8, code 0

Design experiments that will produce ICMP messages of the above type in a packet trace. Which commands did you use to generate the above ICMP message types?

Q8.4)

The Colonel saw the IP packet format and is particularly fascinated by the fragmentation fields and wants to see it in action. Design an experiment where (i)) No fragmentation occurs and (ii) Fragmentation occurs.

Guidance:

- "sendUDP.c" is a simple socket program (will be provided in department portal) that generates a single IP packet of a given size and sends it to the specified destination IP address. Compile the program and design your experiment around it.

(i) Which destination IP address should you use? Try both an existing host and non-existent host within subnet and see what happens.

(ii) Specify the exact commands used to conduct the experiment.

(iii) For the case when no fragmentation happened, note down the values corresponding to the following fields: Total length, Identifier, flags and fragment offset. Why is the total length field so?

(iv) For the case when fragmentation happened, for each fragment, note down the values corresponding to the following fields: Total length, Identifier, flags and fragment offset. Do the values make sense based on what was covered in theory?

Q8.5)

Study of Dynamic Host Configuration Protocol (DHCP)

1) The Colonel is equally fascinated by how hosts obtain IP address and wants to look at the message exchange of this process. One of his staff already procured such a trace, help him interpret the trace.

Guidance:

1. Configuring IP addresses requires root permission.
2. Since you do not have these privileges, for this exercise you will have to make do with a generated trace file "dhcp.out". (Will be provided in Department portal)
3. This trace file was generated using tcpdump and running "dhclient eth0" on a terminal with root permissions. Explore it via wireshark.

(i) What is the IP address of the DHCP server and on what port is it listening on?

(ii) Was any DHCP relay involved in forwarding the DHCP packets? How did you determine the answer?

(iii) When the DHCP server replied, which IP address did it reply to? And why?

(iv) What is the offered IP address to the client and for how long is this address valid?

(v) Apart from the IP address, what additional information has the client received from the dhcp server?

Q8.6)

Capture and study wireless frames that are exchanged between a wireless host and the access point. (Wireless Interface needed in host to perform this experiment.)

Additional Exercises:

Q8.7)

In this exercise, we will use tcpdump to capture a packet containing the link, IP, and TCP headers and use ethereal to analyze this packet. First, run tcpdump -enx -w exe8_3.out you will not see any tcpdump output, since the -w option is used to write the output to the exe8_1.out file. Then, you may want to run ssh to a remote host 6 to generate some TCP traffic. After you login the remote machine, terminate the telnet session and terminate the tcpdump program. Next, you will use wireshark to open the packet trace captured by tcpdump and analyze the captured packets. To do this, run *wireshark -r exe8_3.out &*. The wireshark Graphical User Interface (GUI) will pop up and the packets captured by tcpdump will be displayed. Draw the format of the packet you saved, including the link, IP, and TCP headers, and identify the value of each field in these headers. Express the values in the decimal format. What is the value of the protocol field in the IP header of the packet you saved? What is the use of the protocol field?

Q8.8)

Perform the experiments of Q8.1 and 8.2 using tcpdump.

[OBSERVATION SPACE-LAB 8]

[OBSERVATION SPACE-LAB 8]

[OBSERVATION SPACE-LAB 8]

[OBSERVATION SPACE-LAB 8]

[OBSERVATION SPACE-LAB 8]

[OBSERVATION SPACE-LAB 8]

[OBSERVATION SPACE-LAB 8]

[OBSERVATION SPACE-LAB 8]

LAB NO: 9 **Date :**

DATA LINK LAYER PROTOCOLS

Objectives:

- Understand data link layer protocols
- Observe how data is transferred between network nodes

.

Prerequisites:

- Knowledge on Data Link Layer protocol.

Guidance:

- **ARP Packet**

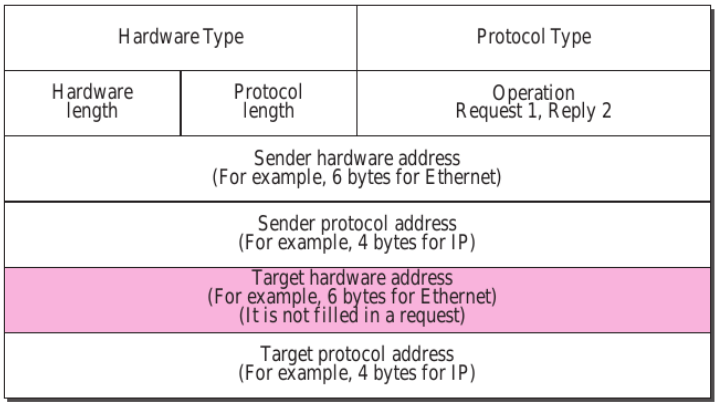


Fig. 9.1 ARP Packet

- **Encapsulation of ARP Packet**

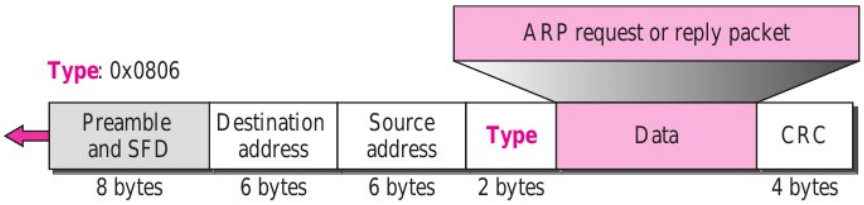


Fig. 9.2 Encapsulation of ARP Packet

Lab exercise:

Q9.1)

Examine the contents of a frame sent by the data-link layer. Find the value of different fields such as destination and source MAC addresses, the value of CRC, the value of the protocol field which shows which payload is being carried by the frame and so on.

Q9.2)

Use arp -a to see the entire ARP table. Observe that all the IP addresses displayed are on the same subnet. If you find that all the remote hosts are in your host's ARP table, you need to delete a remote host (not your workstation) from the table, using, *arp -d* remote_host Save the ARP table for your lab report. While ***tcpdump - enx -w exe9_2.out*** is running, ping a remote host that has no entry in your host ARP table. Then terminate the tcpdump program. Next, run ***wireshark -r exe9_2.out&*** to load the tcpdump trace file. Observe the first few lines of the packet trace to see how ARP is used to resolve an IP address. Run ***arp -a*** to see a new line added in your host's ARP table. Save the new ARP table for your lab report. Note down ARP Request packet and ARP Reply Packet in wireshark window. From the saved tcpdump output, explain how ARP operates. Draw the format of a captured, ARP request and reply including each field and the value. Your report should include the answers for the following questions.

(i) What is the target IP address in the ARP request?

(ii) At the MAC layer, what is the destination Ethernet address of the frame carrying the ARP request?

(iii) What is the frame type field in the Ethernet frame?

(iv) Who sends the ARP reply?

Q9.3)

Examine the content of an ARP packet. Capture the ARP request and ARP reply packets. The interesting fields to examine are those that show what type of source and destination addresses are used in each packet.

Additional Exercises:

Q9.4)

When Col. Raaja read up on how packet forwarding is done at a host, he came to know that in cases where the destination IP address belongs to the host's own subnet, the packet goes directly, otherwise it goes via a router. He wants to check this out. Can you design an experiment that illustrates this? Also, he wants to know what happens if you try to send a packet to "a non-existent host within the same subnet". Help him with that as well.

.

Guidance:

1. As the name of the exercise suggests, this experiment is about exploring the ARP protocol in the context of forwarding. You do not have permissions to delete or set the arp entries but you should be able to view the arp entries currently in the cache.
2. In all cases, you do NOT want the arp entry of the target in the cache. Ensure that this is the case.
3. When you are sending packets to a 'nonexistent host within the same subnet', do wait for at least 10 sec before closing your packet capture tool. The underlying action in this case, takes time.
4. “Within subnet” -- hosts within lab (e.g. 172.16.59.* but not all may be reachable, so check your neighborhood IP addresses). “Outside subnet” -- hosts outside lab (e.g. 10.109.1.11; 10.107.1.1; 10.129.50.11; 10.129.50.12; 10.104.1.1). Nonexistent hosts (e.g. 10.105.12.1 to 10.105.12.7)
   [Note: Above IP addresses may change depending upon configurations of Institute Network]

   (i) For the first case, where you sent packets to a host within the same subnet, specify the command sequence used. How many arp messages were exchanged? Does the arp entry correspond to the remote host IP address you contacted? Explain your observations.
   (ii) For the second case, where you sent packets to a remote host outside the same subnet, specify the command sequence used. How many arp messages were exchanged? Does the *arp* entry correspond to the remote host IP address you contacted? Explain your observations.
   (iii) For the third case where you sent packets to a non-existing host, specify the command sequence used. How many ARP attempts were made to resolve the non-existing IP address? After what time did ARP give up? Hint: In this case -c option of ping is important to use.

Q9.5)

Col. Raaja read that ARP is a good candidate for spoofing. An intruder can generate a gratuitous ARP advertising his MAC address and someone else's IP address and capture the some one's traffic. Given its relevance to security, he wants to know more about this aspect of ARP. Design an experiment to capture *gratuitous ARPs.*

Guidance:

1. 'arping' is a tool that generates gratuitous ARPs.
2. Note that you cannot spoof another person's IP address via ARPing. The goal of this experiment is not to spoof, but to understand gratuitous ARPs. So generate a gratuitous ARP with your own IP address.
3. Gratuitous ARPs can be sent as both ARP requests and ARP replies. Capture both types of packets via appropriate arguments. For requests: e.g. arping -I eth0 -c 5 your_ip_address.
   For replies: e.g. arping -A -I eth0 -c 5 your_ip_address
   (i) How is it ensured that the gratuitous ARPs reach all hosts within the physical network?

(ii) What difference did you observe when gratuitous ARP was sent as a request versus it being sent as a reply

[OBSERVATION SPACE – LAB 9]

[OBSERVATION SPACE – LAB 9]

[OBSERVATION SPACE – LAB 9]

[OBSERVATION SPACE – LAB 9]

[OBSERVATION SPACE – LAB 9]

LAB NO: 10 **Date :**

HANDS ON EXPERIMENTS -1 : LAN & NETWORK SERVICES

(Experiments to be conducted in groups)

Objectives:

- To Understand Design and Deployment of Local Area Networks.
- To provide FTP service over the LAN.

Prerequisites:

- Knowledge ofIPv4 Addressing.
- Basic Knowledge Network Switches and Wireless Access Point.

Lab exercise:

Note : Indicate steps and Commands used with all of these experiments clearly.

Q10.1)

Design and Configure a LAN using a Switch and Wireless Access Point. Assign **static IP** Addresses to all hosts. Check the connectivity between hosts.

Q10.2)

Design and Configure a LAN using a Switch and Wireless Access Point. Assign dynamic **IP** Addresses to all hosts by configuring a DHCP server for your LAN. Check the connectivity between hosts.

Q10.3)

Configure a FTP server in LAN.

[OBSERVATION SPACE-LAB 10]

[OBSERVATION SPACE-LAB 10**]**

[OBSERVATION SPACE-LAB 10**]**

LAB NO: 11 **Date :**

HANDS ON EXPERIMENTS -2 : INTERNETWORKING &NETWORK SERVICES

(Experiments to be conducted in groups)

Objectives:

- To Understand Design and Deployment of Computer Networks
- To provide Network Service over the Computer Networks.

Prerequisites:

- Knowledge ofIPv4 Addressing.
- Basic Knowledge Network Routers, Switches and Wireless Access Point.

Lab exercises

Note : Indicate steps and Commands used with all of these experiments clearly.

Q11.1)

Design and Configure Two LANs using a Switch, Wireless Access Point and a Router.

LAN1: Assign Class **A static IP** Addresses to all hosts.

LAN2: Assign Class **B static IP** Addresses to all hosts.

Interconnect two LANs with a Router. Check the connectivity between two LANs

Q11.2)

Configure a *TELNET/ SSH* server in LAN1 and *Apache Web server* in LAN2 .

Ensure that all services are available to all hosts in both LANS.

[OBSERVATION SPACE-LAB 11**]**

[OBSERVATION SPACE-LAB 11**]**

LAB NO: 12 **Date :**

MINI PROJECT DEMONSTRATION & EVALUATION

Evaluation Criteria:
1. Demonstration
2. Coding
3. PPT Presentation.
4. Q&A

REFERENCES:

1. Behrouz A. Forozan & Firouz Mosharraf,"Computer Networks- A Top Down Approach ", McGraw Hill Education,2011.
2. Neil Matthew, Richard Stones,"Biginning Linux Programming", 4th Edition. Wiley Publishing,Inc,Apr 2011.
3. W.Richard Stevens, Steven A. Rago,"Advanced Programming in the Unix Environment",Third Edition,Addison-Wesley Professional Computing, May 2013.
4. W. Richard Stevens," UNIX Network Programming, Volume 1: The Sockets Networking API", Third Edition,Addison-Wesley Professional Computing, 2003.
5. W. Richard Stevens," UNIX Network Programming, Volume 2, Second Edition: Interprocess Communications, Prentice Hall, 2002.
6. A tcpdump Primer With Examples-Daniel Miessler. [Online]. Available: https://danielmiessler.com/study/tcpdump/ [Accessed: 25- June- 2016] .

7. Wireshark Primer-Instructables. [Online]. Available:

www.instructables.com/id/Wireshark-primer/ [Accessed: 25- June- 2016] .

8. Wireshark-The Easy Tutorial-Filters-OpenManiak.com. [Online]. Available: http:// openmaniak.com/wireshark_filters.php [Accessed: 25- June- 2016] .
9. Introduction and Reference Guide to Wireshark. [Online] . Available: https://thepracticalsysadmin.com/wireshark-reference-guide/ [Accessed: 25- June- 2016] .