Subject: Advanced Computer Networking
Subject Code: MCA 206
Semester IV (2013-16)
Instructor: Dr. Rekha Kashyap

Laboratory Assignments

The aims of the Practical Laboratories prepared for the subject is to enable students:

- Understand the capabilities, limitations and current developments in computer networking.
- Adapt to changes in networking hardware and/or software technology,
- To improve the students' research skills and confidence.
- To require students to read and process on-line manuals, Request For Comment documents (RFCs), Frequently Asked Questions (FAQs), product information etc.

Assignment 1
Submission date: 3/4 Feb 2015

Network Diagnostic Utilities

A successful network administrator r needs tools to diagnose many problems that a network may suffer. Some basic network diagnostic tools (ping, traceroute, tcpdurnp etc.) are introduced in this exercise. Students are required to delve further into the operations of commands and read the relevant RFC of the protocols used in these utilities.

- **Tcpdump** should be used to examine the operations of ARP and traceroute.
- Use the **ping** program to test your own computer(loopback) ,test the host inside the united states/India and and host outside the United States/India
- Use **traceroute or tracert** to find the route from your computer to another computer in your college
- Sow how you can find RTT(Round-trip time) between two routers.
- **Ipconfig** is a Console Command which can be issued to the Command Line Interpreter (or command prompt) to display the network settings currently assigned to any or all network adapters in the machine. This command can be utilised to verify a network connection as well as to verify your network settings.
- **Netstat** displays active TCP connections, ports on which the computer is listening, Ethernet statistics, the IP routing table, IPv4 statistics (for the IP, ICMP, TCP, and UDP protocols), and IPv6 statistics (for the IPv6, ICMPv6, TCP over IPv6, and UDP over IPv6 protocols). Used without parameters, netstat displays active TCP connections.
- The **tracert** command is used to visually see a network packet being sent and received and the amount of hops required for that packet to get to its destination.
- **Pathping** provides information about network latency and network loss at intermediate hops between a source and destination. Pathping sends multiple Echo Request messages to each router between a source and destination over a period of time and then computes results based on the packets returned from each router.
- **telnet** is software that allows users to remotely access another computer such as a server, network device, or other computer. With telnet users can connect to a device or computer, manage a network device, setup a device, transfer files, etc.
- **route** is used to manually configure the routes in the routing table.
- **Arp** displays, adds, and removes arp information from network devices.
- **Nslookup** displays information that you can use to diagnose Domain Name System (DNS) infrastructure. Before using this tool, you should be familiar with how DNS works. The Nslookup command-line tool is available only if you have installed the TCP/IP protocol.
- nbtstat displays protocol statistics and current TCP/IP connections using NBT.

- Netsh : One common way of using netsh is to reset the TCP/IP in Windows 2k/XP Type this in Run or DOS Window – "netsh int ip reset"

  In Windows XP you can run a graphical diagnostics by typing "netsh diag gui" into the run dialogue box. (This may take a little time to startup)

- **Getmac** DOS command is used to show both local and remote MAC addresses. When run with no parameters (ie. getmac) it displays MAC addresses for the local system. When run with the /s parameter (eg. getmac /s \\foo) it displays MAC addresses for the remote computer. When the /v parameter is used, it also displays the associated connection name and network adapter name.
- Find All Active/Used IP Addresses on Your Network
  There is a really neat way that you can quite easily find all active/used IP Addresses on your network without the need for any third party applications or worse, pinging each IP Address individually.

  Open the Command Prompt and type in the following:
  FOR /L %i IN (1,1,254) DO ping -n 1 192.168.10.%i | FIND /i "Reply">>c:\ipaddresses.txt
  Change 192.168.10 to match you own network.

  - Learn System administration (files, processes,
  - permissions, installation)file access permissions, access control &
  - inheritance

Assignment 2
Submission date: 17/18 Feb 2015

Learning the working of various layers in TCP/IP protocol suite using Network Simulator(NET SIM)

Students should install any network Simulator preferably NetSim. They are further supposed to implement laboratory assignments given in the Network Simulator.

- Key network concepts to understand:

packet, protocol, addressing (IP address, domain name, DHCP, DNS, ports), gateway, routers.

Assignment 3
Submission date: 10/11 March 2015

Interprocess Communication and Client Server Programming using Sockets

Programming with sockets gives students experience with protocols and communication between processes without having to deal with machine dependencies at lower levels. Students are required to implement several client-server applications such as simple FTP client and server and a simple multi-user chat client and server as discussed in class Students may use any programming language to develop the application, but, typically, Java is the language of choice, as it is the easiest with which to develop applications for the Internet.

Students are also required to understand interprocess communication by implementing RMI(Remote method invocation). race conditions, concurrent processes, mutual exclusion and other details of interprocess communication must be considered.

Key network concepts to understand :
Internet HTTP, TCP/IP, UDP, NAT, ISP
SMTP protocol, mail formats and MIME,
mail access protocols (POP and IMAP)

Assignment 4
Submission date: 24/25 March 2015

Implementation of Security Protocols

The importance of network security cannot be underestimated. Learning available tools for the prevention and management of security breaches is part of this assignment. The Students are further required to understand and implement various encryption algorithms used in SSL, TLS and IpSec. Following excersices should be implemented.

- Write a program that calculates the message digest of a txt using MD5 algorithm.
- Write a program to implement RSA algorithm.
- Write a program in Java, which performs a digital signature on a given text.
- Can we create a certificate programmatically in Java or .NET. Try doing it.
- Investigate how to use SSL in Java and .NET. Write an SSL client and server in these technologies.

a) Ceasar Cipher Implementation

The transformation can be represented by aligning two alphabets; the cipher alphabet is the plain alphabet rotated left or right by some number of positions. For instance, here is a Caesar cipher using a left rotation of three places, equivalent to a right shift of 23 (the shift parameter is used as the key):

Plain: ABCDEFGHIJKLMNOPQRSTUVWXYZ
Cipher: XYZABCDEFGHIJKLMNOPQRSTUVW

When encrypting, a person looks up each letter of the message in the "plain" line and writes down the corresponding letter in the "cipher" line. Deciphering is done in reverse, with a right shift of 3.

Ciphertext: QEB NRFZH YOLTK CLU GRJMP LSBO QEB IXWV ALD
Plaintext: the quick brown fox jumps over the lazy dog

The encryption can also be represented using modular arithmetic by first transforming the letters into numbers, according to the scheme, A = 0, B = 1,..., Z = 25. Encryption of a letter x by a shift n can be described mathematically as,

$$E_n(x) = (x + n) \mod 26.$$

Decryption is performed similarly,

$$D_n(x) = (x - n) \mod 26.$$

The replacement remains the same throughout the message, so the cipher is classed as a type of *monoalphabetic substitution*, as opposed to *polyalphabetic substitution*.

b) Diffie Hellman Key Exchange (Implement the following algorithm)

Alice and Bob agree on two values: a large prime number p, and a generator g, 1 < g < p. (g is a primitive root of p)(3,2 / 5,2 / 9,2 / 11,2 / 13,6….)

These values are known to everyone.

In secret, Alice picks a value a, with 1 < a < p.
In secret, Bob picks a value b, with 1 < b < p.

Alice calculates g^a (mod p), call this f(a) and sends it to Bob.
Bob calculates g^b (mod p), call this f(b) and sends it to Alice.

Note that f(a) and f(b) are also known by everyone.

In secret, Alice computes $f(b)^a \pmod p$ – this is the exchanged key.
In secret, Bob computes $f(a)^b \pmod p$ – this is again, the exchanged key.

Why does this work?

$f(b)^a \equiv (g^b)^a \equiv g^{ab} \pmod p$. Similarly,
$f(a)^b \equiv (g^a)^b \equiv g^{ab} \pmod p$.

Here's a concrete example:

Let p = 37 and g = 13.

Let Alice pick a = 10. Alice calculates $13^{10} \pmod{37}$ which is 4 and sends that to Bob.

Let Bob pick b = 7. Bob calculates $13^7 \pmod{27}$ which is 32 and sends that to Alice.
(Note: 6 and 7 are secret to Alice and Bob, respectively, but both 4 and 32 are known by all.)
Alice receives 32 and calculates $32^{10} \pmod{37}$ which is 30, the secret key.
Bob receives 4 and calculates $4^7 \pmod{37}$ which is 30, the same secret key.

Note that neither Alice nor Bob chose 30, but that they ended up with that secret key anyway. Furthermore, note that even with knowing p = 37, g = 13, f(a) = 4 and f(b) = 32, it is difficult to ascertain the secret key, 30 without doing a brute force check.

c) Implement simplest hash function: Bitwise XOR of every block(Known as longitudinal redundancy check)
All hash functions operate using the following general principles:
a) The input string is viewed as a sequence of n-byte blocks.

b) The input is processed one block at a time in an iterative fashion to produce an n-bit hash function.

The simplest hash function is the list-by-list XOR of every block, expressed as following:

$$C_i = b_{i1} \oplus b_{i2} \oplus \cdots \oplus b_{im}$$

- C_i = i-th bit of the hash code, $1 \le i \le n$
- m = number of n-bit blocks in the input
- b_{ij} = i-th bit in j-th block

bit 1	bit 2	….	bit n
b11	b21		bn1
b12			bn2
:			
:			
b1m			bnm
C1	C2		Cn

(Live Project)

Submission date: 13/14 Apr 2015

Students should implement a project based on the networking concepts with proper software lifecycle and documentation. Suggested topics are

- Simulation of various Routing algorithms used at the Network layer.
- Simulation of Flow control algorithms at Transport layer.
- Application layer protocols Implementation (RFC should be strictly followed).
- DHCP Implementation(RFC should be strictly followed).
- Implementation of Multicasting Routing Protocols(RFC should be strictly followed).
- Security Protocols Implementation.
- Mobile IP Applications.
- Mobile Applications.