|  |  |  |  |
| --- | --- | --- | --- |
| **Raj Kumar Goel Institute of Technology, Ghaziabad**    **COMPUTER NETWORKS LABFILE(KCS653)** | | | |
|  | **Name** |  |  |
| **Roll No.** |  |
| **Section-Batch** |  |
|  | | | |

**VISION OF THE INSTITUTE**

To continually develop excellent professionals capable of providing sustainable solutions to challenging problems in their fields and prove responsible global citizens.

**MISSION OF THE INSTITUTE**

We wish to serve the nation by becoming a reputed deemed university for providing value based professional education.

**VISION OF THE DEPARTMENT**

To be recognized globally for delivering high quality education in the ever-changing field of computer science & engineering, both of value & relevance to the communities we serve.

**MISSION OF THE DEPARTMENT**

1. To provide quality education in both the theoretical and applied foundations of Computer Science and train students to effectively apply this education to solve real worldproblems.

2. To amplify their potential for lifelong high-quality careers and give them a competitive advantage in the challenging global work environment.

**PROGRAM EDUCATIONAL OUTCOMES (PEOs)**

PEO 1: Learning: Our graduates to be competent with sound knowledge in field of Computer Science &Engineering.

PEO 2: Employable: To develop the ability among students to synthesize data and technical concepts for application to software product design for successful careers that meet the needs of Indian and multinational companies.

PEO 3: Innovative: To develop research oriented analytical ability among students to prepare them for making technical contribution to the society.

PEO 4: Entrepreneur / Contribution: To develop excellent leadership quality among students which they can use at different levels according to their experience and contribute for progress and development in the society.

**PROGRAM OUTCOMES (POs)**

**Engineering Graduates will be able to:**

**PO1: Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PO2:** | **Problem** | | **analysis**: Identify, formulate, review research literature, and | | | | | | | | | | | |
| analyze | complex | | engineering problems | | | | | reaching | | substantiated conclusions using | | | | |
| first principles of mathematics, | | | | | natural sciences, and engineering | | | | | | | Sciences. | |  |
| **PO3: Design/development of solutions**: | | | | | | | | | Design | | solutions | | for | complex |
| engineering | | problems | | and | design | system components | | | | | or processes that meet the | | | |
| specified needs | | | with | appropriate | | consideration | | | for the public health and safety, and | | | | | |
| the cultural, societal, and environmental | | | | | | | Considerations. | | | |  |  |  |  |
| **PO4:** | **Conduct** | | **investigations** | | | **of** | **complex** | | **problems**: Use research-based | | | | | |
| knowledge | | and | research | | methods | | including | | design of experiments, analysis and | | | | | |
| interpretation of data, and synthesis of the | | | | | | | | information | | to provide valid conclusions. | | | | |
| **PO5:** | **Modern** | | **tool usage**: Create, select, | | | | | | and | apply | | appropriate | | techniques, |
| resources, a n d modern | | | | engineering and IT tools including | | | | | | | prediction | | and modelling to | |
| complex engineering activities with an | | | | | | | Understanding of the limitations. | | | | | |  |  |

**PO6: The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

**PO7: Environment and sustainability**: Understand the impact of the professional

Engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

**PO8: Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

**PO9: Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

**PO10: Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to Comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

**PO11: Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

**PO12: Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

**PROGRAM SPECIFIC OUTCOMES (PSOs)**

PSO1: The ability to use standard practices and suitable programming environment to develop software solutions.

PSO2: The ability to employ latest computer languages and platforms in creating innovative career opportunities.

.

.

**COURSE OUTCOMES (COs)**

|  |  |
| --- | --- |
| C321.1 | Simulate different network topologies |
|  |  |
|  |  |
| C321.2 | Implement various framing networks for data link layer |
|  |  |
|  |  |
| C321.3 | Implement various Error and Flow control techniques |
|  |
|  |
|  |  |
| C321.4 | Implement network routing and addressing techniques |
|  |  |
| C321.5 | Implement transport and security mechanism |
|  |  |

**CO-PO MAPPING**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CO** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** |
| **C321.1** | 3 | 3 |  |  |  |  | 1 |  | 1 | 2 |  | 1 | 1 |  |
| **C321.2** | 3 | 3 | 2 | 1 | 1 |  | 1 |  | 1 |  |  | 2 | 2 | 1 |
| **C321.3** | 3 | 3 | 3 | 1 | 1 | 1 | 1 |  | 2 |  | 1 | 2 | 2 | 1 |
| **C321.4** | 3 | 3 | 3 | 2 | 1 |  |  |  | 2 |  | 1 | 2 | 2 | 1 |
| **C321.5** | 3 | 3 | 1 |  |  | 1 | 1 |  | 1 |  |  | 2 | 1 |  |
| **C321** | 3 | 3 | 2.25 | 1.3 | 1 | 1 | 1 |  | 1.4 | 2 | 1 | 1.8 | 1.6 | 1 |

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Experiment No.** | **Experiment**  **Name** | **Dateof**  **Conduction** | **Dateof**  **Submission** | **Faculty**  **Signature** |
| 1 | To learn handling and configuration of networking hardware like RJ-45 connector, CAT-6cable, crimping tool, etc |  |  |  |
| 2 | Aim: Study of following Network Devices in Detail   * + Repeater   + Hub   + Switch   + Bridge   + Router   Gateway |  |  |  |
| 3 | Study of basic network command and Network configuration commands. |  |  |  |
| 4 | Configure a Network topology using packet tracer software. |  |  |  |
| 5 | STUDY OF NETWORK SIMULATOR(NS) |  |  |  |
| 6A | Simulation of Distance Vector/LinkStateRouting |  |  |  |
| 6B | SIMULATIONOFLINKSTATEROUTINGALGORITHM |  |  |  |
| 7 | Implementation of Stop and Wait Protocol and Sliding WindowProtocol. |  |  |  |
| 8 | Study of Socket Programming and Client–Server model |  |  |  |
| 9 | Write a code simulating ARP/RARP protocols. |  |  |  |
| 10A | Write a code simulating PING commands |  |  |  |
| 10B | Write a code simulating Trace route commands |  |  |  |
| 11 | Create a socket for HTTP for web page upload and download. |  |  |  |
| 12 | Write a program to implement RPC(Remote Procedure Call) |  |  |  |
| 13 | Implementation of Sub netting. |  |  |  |
| 14 | IMPLEMENTATION OF TCP/IP ECHO.ALGORITHM |  |  |  |
| 15 | |  | | --- | | Program to implement DNS in java | |  |  |  |

COMPUTER NETWORKS LAB(KCS-653)(CS,VISEM) Page2

**EXPERIMENT-1**

**Aim**: To learn handling and configuration of networking hardware like RJ-45 connector, CAT-6cable, crimping tool, etc.

**Apparatus (Components):** RJ-45connector, Climping Tool, Twisted pair Cable

**Procedure:** To do these practical following steps should be done:

1. Start by stripping off about 2 inches of the plastic jacket off the end of the cable. Be very careful at this point, as to not nick or cut into the wires, which are inside. Doing so could alter the characteristics of your cable, or even worse render is useless. Check the wires, one more time for nicks or cuts. If there are any, just whack the whole end off, and start over.
2. . Spread the wires apart, but be sure to hold onto the base of the jacket with your other hand. You do not want the wires to become untwisted down inside the jacket. Category 5 cable must only have 1/2 of an inch of 'untwisted' wire at the end; otherwise it will be 'out of spec'. At this point, you obviously have ALOT more than 1/2 of an inch of un-twisted wire..

You have 2 end jacks, which must be installed on your cable. If you are using a pre-made cable, with one of the ends whacked off, you only have one end to install - the crossed over end. Below are two diagrams, which show how you need to arrange the cables for each type of cable end. Decide at this point which end you are making and examine the associated picture below.

**Diagram shows you how to prepare Cross wired connection**



**Diagram shows you how to prepare straight through wired connection**



**EXPERIMENT- 2**

**Aim: Study of following Network Devices in Detail**

* + **Repeater**
  + **Hub**
  + **Switch**
  + **Bridge**
  + **Router**
  + **Gateway**

**Apparatus (Software):** No software or hardware needed.

**Procedure:** Following should be done to understand this practical.

1. **Repeater:** Functioning at Physical Layer. A **repeater** is an electronic device that receives a signal and retransmits it at a higher level and/or higher power or onto the other side of an obstruction so that the signal can cover longer distances. Repeater have two ports, so can not be use to connect for more than two devices
2. **Hub:**An**Ethernethub**,**activehub**,**networkhub**,**repeaterhub**,**hub**or**concentrator**

is a device for connecting multiple twisted pair or fiber optic Ethernet devices together andmaking them act as a single network segment. Hubs work at the physical layer (layer 1) of theOSI model. The device is a form of multiport repeater. Repeater hubs also participate incollisiondetection,forwardingajamsignaltoallportsifitdetectsacollision.

1. **Switch:**A**network switch** or **switching hub** is a computer networking device thatconnects network segments.The term commonly refers to a network bridge that processes androutes data at the data link layer (layer 2) of the OSI model. Switches that additionallyprocessdataatthenetworklayer(layer3andabove)areoftenreferredtoasLayer3switchesor multilayerswitches.
2. **Bridge:** A **network bridge** connects multiple network segments at the data link layer(Layer 2) of the OSI model. In Ethernet networks, the term *bridge* formally means a devicethat behaves according to the IEEE 802.1D standard. A bridge and switch are very muchalike; a switch being a bridge with numerous ports. *Switch* or *Layer 2 switch* is often usedinterchangeably with *bridge*.Bridges can analyze incoming data packets to determine if thebridgeisabletosendthegivenpackettoanothersegmentofthenetwork.
3. **Router:** A **router** is an electronic device that interconnects two or more computernetworks, and selectively interchanges packets of data between them. Each data packetcontains address information that a router can use to determine if the source and destinationare on the same network, or if the data packet must be transferred from one network toanother. Where multiple routers are used in a large collection of interconnected networks, theroutersexchangeinformationabouttargetsystemaddresses,sothateachroutercanbuildupatableshowingthepreferredpathsbetweenanytwosystemsontheinterconnectednetworks.
4. **GateWay:**Inacommunicationsnetwork,anetworknodeequippedforinterfacingwith

anothernetworkthatusesdifferentprotocols.

* + A gateway may contain devices such as protocol translators, impedance matchingdevices, rate converters, fault isolators, or signal translators as necessary to providesystem interoperability. It also requires the establishment of mutually acceptableadministrativeproceduresbetweenbothnetworks.
  + Aprotocoltranslation/mappinggatewayinterconnectsnetworkswithdifferentnetworkprotocoltechnologiesbyperformingtherequiredprotocolconversions.

**EXPERIMENT-3**

**Aim:** Study of basic network command and Network configuration commands.

**Apparatus (Software):** Command Prompt and Packet Tracer.

**Procedure:** To do this EXPERIMENT-follow these steps:

In this EXPERIMENT-studentshavetounderstandbasicnetworkingcommandse.gping,tracertetc.

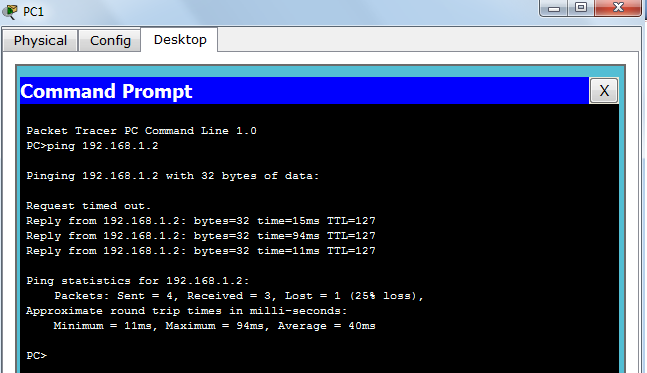
All commands related to Network configuration which includes how to switch to privilege mode and normal mode and how to configure router interface and how to save this configuration to flash memory or permanent memory.

This commands includes

* + Configuring the Router commands
  + General Commands to configure network
  + Privileged Mode commands of a router
  + Router Processes & Statistics
  + IP Commands
  + Other IP Commands e.g. show ip route etc.

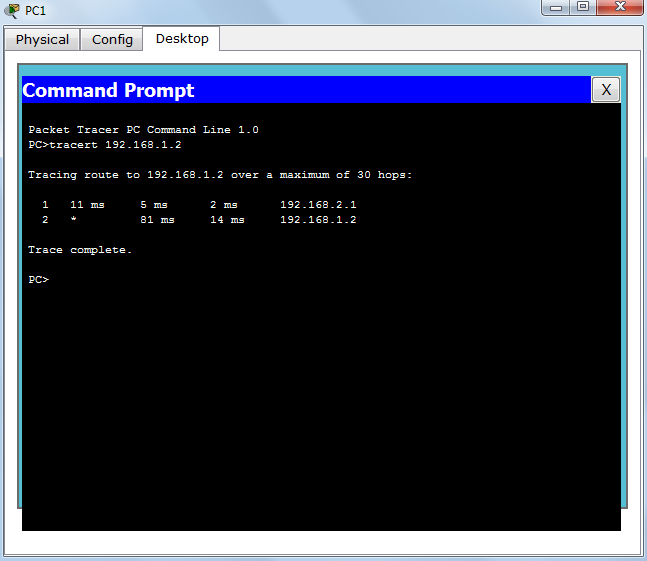
**Ping:**

Ping (8) sends an ICMPECHO\_REQUEST packet to the specified host. If the host responds, you get an ICMP packet back. Sound strange? Well, you can “ping” an IP address to see if a machine is alive .If there is no response, you know something is wrong.



**Trace route:**

Tracert is a command which can show you the path a packet of information takes from your computer to one you specify. It will list all the routers it passes through until it reaches its destination ,or fails to an dis discarded.Inadditiontothis,itwilltellyouhowlongeach'hop'fromroutertorouter takes.



**nslookup:**

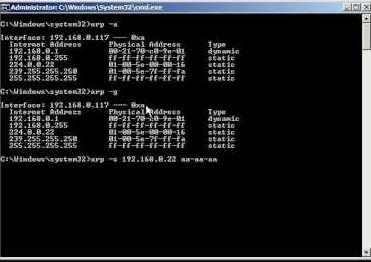
DisplaysinformationfromDomainNameSystem(DNS)nameservers.

NOTE:Ifyouwritethecommandasaboveitshowsasdefaultyourpc'sservernamefirstly.

**ARP:**

TheARPcommands toview, display,ormodifythedetails/information inan ARPtable/cache.

The ARP cache or table has the dynamic list of IP and MAC addresses of those devices to whichyourcomputerhascommunicatedrecentlyinalocalnetwork.Thepurposeofmaintainingan ARP tableis that when you want to communicate with another device, your device does notneedto send theARP requestforthe MACaddress of that device.



**TELNET:**

Telnet is an application that is used to connect to a remote host’s command line terminalinterface. Network and system administrators use this application to configure and administernetworkdevicessuchasservers,routers,switches,etc.Thisapplicationisbasedontheconnection-orientedTransmission ControlProtocol(TCP).

## FTP:

FTP is the simplest file transfer protocol to exchange files to and from a remote computer ornetwork. Similar to Windows, Linux and UNIX operating systems also have built-in command-lineprompts that can beusedas FTP clients to makeanFTPconnection.

Inanycommandmode,youcangetalistofavailablecommandsbyenteringaquestionmark(?).

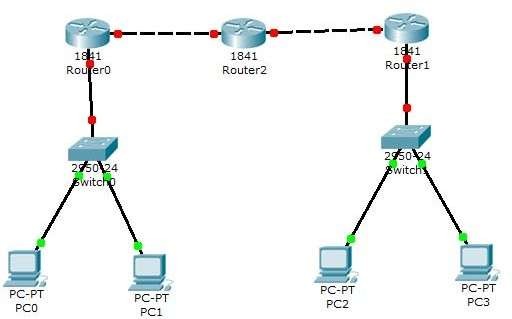
**EXPERIMENT- 4**

**Aim:** Configure a Network topology using packet tracer software.

**Apparatus(Software):**Packet tracer Software

**Procedure:**Toimplementthispracticalfollowingnetworktopologyisrequiredtobeconfiguredusingthecommandslearnedinpreviouspractical.

Afterconfiguringthegivennetworkapacketshouldbepingfromanyonemachinetoanother.



## Router0ConfigurationCommand:.........

Continuewithconfigurationdialog?[yes/no]:no

PressRETURNtogetstarted!Router>

Router>EnableRouter#configt

Enterconfigurationcommands,oneperline.EndwithCNTL/Z.

Router(config)#hostnamerouter0router0(config)#interfacefastethernet0/0

router0(config-if)#ipaddress192.168.1.1255.255.255.0router0(config-if)#descriptionrouter0fastethernet0/0router0(config-if)#noshutdown

%LINK-5-CHANGED:InterfaceFastEthernet0/0,changedstatetouprouter0(config-if)#exit

router0(config)#interfacefastethernet0/1router0(config-if)#descriptionrouter0fastethernet0/1router0(config-if)#noshutdown

%LINK-5-CHANGED:InterfaceFastEthernet0/1,changedstatetouprouter0(config-if)#exit

router0(config)#exit

%SYS-5-CONFIG\_I:Configuredfromconsolebyconsolerouter0#showrunning-config

Buildingconfiguration...

Currentconfiguration:437bytes

!

version12.4

noservicepassword-encryption

!

hostnamerouter0

!

!

!

!

!

ipsshversion1

!

!

interfaceFastEthernet0/0descriptionrouter0fastethernet0/0

ipaddress192.168.1.1255.255.255.0

duplex autospeedauto

!

interfaceFastEthernet0/1description router0 fastethernet 0/1noipaddress

duplex autospeedauto

!

interfaceVlan1noipaddress

shutdown

!

ipclassless

!

!

!

!

!

linecon0

linevty04login

!

!

endrouter0#

router0#

router0#copyrunning-configstartup-configDestinationfilename[startup-config]?

Building configuration...[OK]

router0#

### EXPERIMENT5

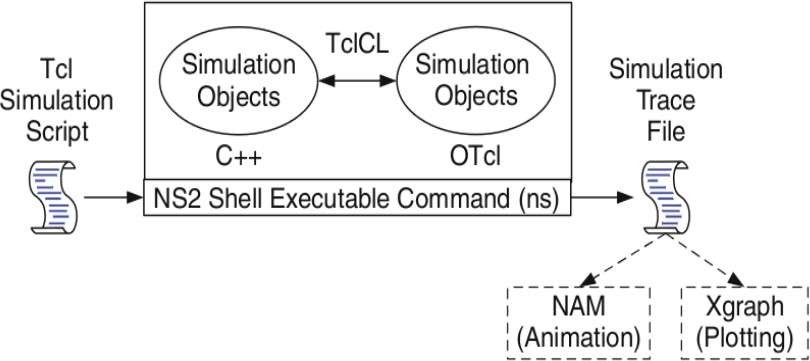
**AIM: STUDY OF NETWORK SIMULATOR(NS)**

### THEORY:

Network Simulator (Version 2), widely known as NS2, is simply an event driven simulationtoolthathas provedusefulin studyingthe dynamicnatureofcommunicationnetworks.

Simulation of wired as well as wireless network functions and protocols (e.g., routingalgorithms, TCP, UDP) can be done using NS2. In general, NS2 provides users with a way ofspecifying such network protocols and simulating their corresponding behaviors. Due to itsflexibility and modular nature, NS2 has gained constant popularity in the networking researchcommunitysinceits birth in 1989.

### BasicArchitectureof NS2:



The above figure shows the basic architecture of NS2. NS2 provides users with anexecutable command ns which takes on input argument, the name of a Tcl simulationscripting file. Users are feeding the name of a Tcl simulation script (which sets up asimulation)as an input argumentofan NS2 executablecommand ns.

Inmostcases,asimulationtracefileiscreated,andisusedtoplotgraphand/ortocreateanimation. NS2 consists of two key languages: C++ and Object-oriented Tool CommandLanguage(OTcl).WhiletheC++definestheinternalmechanism(i.e.,abackend)ofthe

simulationobjects,theOTclsetsupsimulationbyassemblingandconfiguringtheobjectsaswell as schedulingdiscreteevents (i.e.,afrontend).

The C++ and the OTcl are linked together using TclCL. Mapped to a C++ object,variablesintheOTcldomainsaresometimesreferredtoashandles.Conceptually,ahandle(e.g., n as a Node handle) is just a string (e.g.,\_o10) in the OTcl domain, and does notcontain any functionality. instead, the functionality (e.g., receiving a packet) is defined inthe mapped C++ object (e.g., of class Connector). In the OTcl domain, a handle acts as afrontendwhichinteractswithusersandotherOTclobjects.Itmay definesitsownproceduresandvariablestofacilitatetheinteraction.Notethatthememberproceduresandvariables in the OTcl domain are called instance procedures (instprocs) andinstancevariables(instvars),respectively

### Tclscripting:

* Tclisageneralpurposescriptinglanguage.[Interpreter]
* Tclrunsonmost ofthe platformssuchasUnix,Windows,and Mac.
* ThestrengthofTclisitssimplicity.
* Itisnotnecessaryto declarea datatypeforvariableprior totheusage

### BasicsofTCL

Syntax:commandarg1arg2 arg3

### Hello World!

puts stdout{Hello, World!}Hello,World!

### Variables

CommandSubstitution

seta5set len[stringlengthfoobar]

setb $aset len [expr[stringlength foobar] +9]

### SimpleArithmetic

expr7.2 / 4

### Procedures

procDiag{a b} {

set c [expr sqrt($a \* $a + $b \* $b)]return$c}

puts ―Diagonal of a 3, 4 right triangle is [Diag 3 4]‖Output: Diagonalofa3,4right triangleis 5.0

### Loops

|  |  |
| --- | --- |
| while{$i<$n} { | for{set i 0}{$i <$n} {incri} |
| . . . | { |
| } | . . . |
|  | } |

**NSSimulatorPreliminaries.**

1. Initializationandterminationaspectsofthenssimulator.
2. Definitionofnetworknodes, links,queuesandtopology.
3. Definitionofagentsandofapplications.
4. Thenamvisualizationtool.
5. Tracingandrandomvariables.

### InitializationandTermination ofTCLScriptinNS-2

Annssimulationstarts with thecommand

### setns[newSimulator]

Which is thus the first line in the tcl script? This line declares a new variable as using the setcommand, you can call this variable as you wish, In general people declares it as ns becauseit is an instance of the Simulator class, so an object the code[new Simulator] is indeed theinstallationoftheclass Simulator usingthe reserved wordnew.

In order to have output files with data on the simulation (trace files) or files used forvisualization(nam files), weneed tocreate thefilesusing―”open”command:

### #Open the Tracefile

**settracefile1 [openout.trw]**

### $ns trace-all $tracefile1#Open the NAMtracefile

**setnamfile[openout.namw]**

### $nsnamtrace-all$namfile

Theabovecreatesadtatracefilecalled―out.tr‖andanamvisualization tracefilecalled

―out.nam‖.Within the tcl script,these files are not called explicitly by their names, butinsteadbypointers that aredeclaredabove and called ―tracefile1and ―nam file

respectively. Remark that they begins with a # symbol.The second line open the file ―out.tr‖to be used for writing, declared with the letter ―w‖.The third line uses a simulator methodcalledtrace-all that have asparameter thenameofthefilewherethetraceswill go.

The last line tells the simulator to record all simulation traces in NAM input format. Italso gives the file name that the trace will be written to later by the command $ns flush-trace.Inourcase,this will bethe filepointedat bythepointer―$namfile ,i.e the file―out.tr‖.

Theterminationoftheprogram isdoneusinga―finish‖procedure.

### #Definea„finish‟procedure

**Procfinish{}{**

### globalnstracefile1 namfile

**$ns flush-traceClose $tracefile1Close$namfile**

### Exec nam out.nam &Exit 0

The word proc declares a procedure in this case called finish and without arguments. Theword global is used to tell that we are using variables declared outside the procedure. Thesimulatormethod―flush-trace”willdumpthetracesontherespectivefiles.Thetclcommand

―close”closesthetracefilesdefinedbeforeandexecexecutesthenamprogramforvisualization.

Thecommandexitwillendstheapplicationandreturnthenumber0asstatustothesystem.

Zero

isthe defaultforacleanexit.Other valuescanbeusedtosaythatisaexitbecausesomethingfails.

Attheendofnsprogramweshouldcalltheprocedure―finish‖andspecifyatwhattimethetermination should occur. For example,

### $nsat125.0“finish”

will be used to call ―finish‖ at time 125sec.Indeed,the at method of the simulator allows usto

scheduleeventsexplicitly.

Thesimulationcanthenbegin usingthecommand

### $ns runDefinitionofanetworkoflinks andnodes

Thewaytodefine anodeis

### setn0[$nsnode]

The node is created which is printed by the variable n0. When we shall refer to that node inthe

scriptweshall thuswrite$n0.

Once we define several nodes, we can define the links that connect them. An example ofadefinition ofalink is:

### $nsduplex-link$n0$n210Mb 10msDropTail

Which means that $n0 and $n2 are connected using a bi-directional link that has 10msof propagation delay and a capacity of 10Mb per sec for each direction. To define adirectionallinkinsteadofabi-directionalone,we shouldreplace “duplexlink”by“simplex-

link”.

In NS, an output queue of a node is implemented as a part of each link whose input isthat node. The definition of the link then includes the way to handle overflow at that queue.Inourcase,if thebuffercapacityoftheoutputqueueis exceededthen the last packetto

arrive is dropped. Many alternative options exist, such as the RED (Random Early Discard)mechanism, the FQ (Fair Queuing), the DRR (Deficit Round Robin), the stochastic FairQueuing(SFQ)andthe CBQ (which includinga priorityandaround-robinscheduler).

In ns, an output queue of a node is implemented as a part of each link whose input is thatnode. We should also define the buffer capacity of the queue related to each link. An examplewould be:

### #setQueueSizeoflink(n0-n2)to 20

**$nsqueue-limit$n0$n220**

### AgentsandApplications

We need to define routing (sources, destinations) the agents (protocols) the applicationthat usethem

### FTPoverTCP

TCP is a dynamic reliable congestion control protocol. It uses Acknowledgements created bythe destination to know whether packets are well received. There are number variants of theTCP protocol, such as Tahoe, Reno, NewReno, Vegas. The type of agent appears in the firstline:

### settcp [new Agent/TCP]

Thecommand**$ns attach-agent$n0$tcp**definesthesourcenodeofthetcpconnection.

Thecommand

### setsink[newAgent/TCPSink]

Definesthebehaviorofthedestination nodeofTCP andassignsto itapointercalledsink

### #SetupaUDPconnection

**setudp[newAgent/UDP]**

### $ns attach-agent $n1 $udpsetnull [newAgent/Null]

**$nsattach-agent$n5$null**

### $nsconnect$udp$null

**$udpsetfid\_2**

### #setupaCBRoverUDPconnection

**set cbr [newApplication/Traffic/CBR]**

### $cbrattach-agent$udp

**$cbrset packetsize\_ 100**

### $cbrset rate\_0.01Mb

**$cbrsetrandom\_false**

Aboveshowsthedefinitionof aCBRapplicationusingaUDPagent.Thecommand

**$nsattach-agent$n4$sink** definesthedestinationnode.Thecommand**$nsconnect$tcp**

$sinkfinallymakes theTCPconnection betweenthesourceanddestination nodes.

TCP has many parameters with initial fixed defaults values that can be changed ifmentioned explicitly. For example, the default TCP packet size has a size of 1000bytes.Thiscan be changed to another value, say 552bytes, using the command $tcp set packetSize\_ 552.When we have several flows, we may wish to distinguish them so that we can identify themwithdifferent colorsin thevisualizationpart.Thisisdonebythecommand $tcpsetfid\_1 thatassigns to the TCP connection a flow identification of“1”.We shall later give the flowidentificationof“2”to theUDP connection.

### CBRoverUDP

A UDP source and destination is defined in a similar way as in the case of TCP. Instead ofdefining the rate in the command $cbr set rate\_ 0.01Mb, one can define the time intervalbetweentransmission ofpacketsusingthe command.

### $cbr set interval\_ 0.005Thepacket sizecan beset to somevalueusing

**$cbrsetpacketSize\_<packetsize>**

### SchedulingEvents

NSisadiscreteeventbasedsimulation.Thetcpscriptdefineswheneventshouldoccur.

The initializing command set ns [new Simulator] creates an event scheduler, and events arethenscheduled usingtheformat:

### $nsat<time><event>

The scheduler is started when running ns that is through the command $ns run. The beginningandendoftheFTP andCBRapplicationcanbedonethroughthe followingcommand

### $nsat0.1“$cbrstart”

**$nsat1.0“$ftpstart”**

### $nsat124.0“$ftpstop”

**$nsat124.5“$cbrstop”**

### RESULT:

ThustheNetworkSimulator2isstudiedindetail.

### EXPERIMENT6

### Simulation of Distance Vector/LinkStateRouting.

.

### Distance Vector routingprotocolROUTINGAim:

TosimulateandstudythelinkstateroutingalgorithmusingsimulationusingNS2.

**DistanceVectorrouting protocol**

Routing is the process of selecting best paths in a network. In the past, the termrouting was also used to mean forwarding network traffic among networks. However thislatter function is muchbetter described as simply forwarding. Routing is performedformany kinds of networks, includingthe telephone network (circuit switching), electronicdata networks (such as the Internet), and transportation networks. This article is concernedprimarilywithroutinginelectronicdatanetworksusingpacketswitchingtechnology

.In packet switching networks, routing directs packet forwarding (the transit of logicallyaddressed network packets from their source toward their ultimate destination) throughintermediate nodes. Intermediate nodes are typically network hardware devices such asrouters,bridges,gateways,firewalls,orswitches.General-purposecomputerscanalsoforward packets and perform routing, though they are not specialized hardware and maysuffer from limited performance.The routing process usually directs forwarding on thebasisofroutingtableswhich maintainarecordoftheroutestovariousnetworkdestinations.Thus, constructing routing tables, which are held in the router's memory, is very importantforefficientrouting.Mostroutingalgorithmsuseonlyonenetworkpathatatime.Multipathroutingtechniquesenabletheuseofmultiplealternativepaths.Incaseofoverlapping/equal routes, the followingelementsareconsideredinordertodecidewhichroutesgetinstalledintotheroutingtable(sortedbypriority):

1. Prefix-Length: where longer subnet masks are preferred (independent of whether it iswithinarouting

protocoloroverdifferentroutingprotocol)

1. Metric: where a lower metric/cost is preferred (only valid within one and the sameroutingprotocol)
2. Administrativedistance:wherealowerdistanceispreferred(onlyvalidbetweendifferentroutingprotocols)Routing,inamorenarrowsenseoftheterm,isoftencontrasted with bridging in its assumption that network addresses are structured and thatsimilar addresses imply proximity within the network. Structured addresses allow a singleroutingtableentrytorepresenttheroutetoagroupofdevices.Inlargenetworks,

structuredaddressing(routing,inthenarrowsense)outperformsunstructuredaddressing(bridging).RoutinghasbecomethedominantformofaddressingontheInternet.Bridgingisstillwidelyusedwithinlocalizedenvironments.

### Algorithm

There are several variants of flooding algorithm. Most work roughlyasfollows:

1. Eachnodeactsasbothatransmitterandareceiver.
2. Each node tries to forward every message to every one of its neighbours except thesource node. This results in every message eventually being delivered toall reachableparts of the network. Algorithms may need to bemorecomplexthanthis,since,insome case, precautions have to betakentoavoid wastedduplicatedeliveriesandinfiniteloops,andtoallowmessagestoeventuallyexpirefromthesystem.Avariantof flooding called selective flooding partially addresses these issues by only sendingpackets to routers in the same direction. In selectivefloodingtheroutersdon'tsendeveryincoming packetonevery linebutonlyonthoselineswhicharegoingapproximatelyintherightdirection.

### Program:

setns[newSimulator]

setnf[openout.namw]

$nsnamtrace-all$nf

settr[openout.trw]

$nstrace-all$tr

procfinish{}{

globalnfnstr

$nsflush-trace

close$tr

execnamout.nam&

exit0

}

set n0 [$ns node]set n1 [$ns node]set n2 [$ns node]setn3[$nsnode]

$nsduplex-link$n0$n110Mb10msDropTail

$nsduplex-link$n1$n310Mb10msDropTail

$nsduplex-link$n2$n110Mb10msDropTail

$nsduplex-link-op$n0$n1orientright-down

$nsduplex-link-op$n1$n3orientright

$nsduplex-link-op$n2$n1 orientright-up

settcp[newAgent/TCP]

$nsattach-agent$n0$tcp

setftp[newApplication/FTP]

$ftpattach-agent$tcp

setsink[newAgent/TCPSink]

$nsattach-agent$n3$sink

setudp[newAgent/UDP]

$nsattach-agent$n2$udp

setcbr[newApplication/Traffic/CBR]

$cbrattach-agent$udp

setnull[newAgent/Null]

$nsattach-agent$n3$null

$nsconnect$tcp$sink

$nsconnect$udp$null

$nsrtmodel-at1.0down$n1$n3

$nsrtmodel-at2.0up$n1$n3

$nsrtprotoDV

$nsat0.0"$ftpstart"

$nsat0.0 "$cbrstart"

$nsat5.0"finish"

$nsrun

**Result:**

ThustheDistanceVectorRoutingAlgorithmwasSimulatedandstudied

### 6b) SIMULATIONOFLINKSTATEROUTINGALGORITHM

**Aim:**

Tosimulateandstudy thelinkstateroutingalgorithmusingsimulationusingNS2.

**LinkStateRoutingprotocol** In link state routing, each router shares its knowledge of its neighborhood with every otherrouterin the

internet work. (i) Knowledge about Neighborhood: Instead of sending its entire routing tablearouter sends

info about its neighborhood only. (ii) To all Routers: each router sends this information toeveryotherrouter

on the internet work not just to its neighbor .It does so by a process called flooding.(iii)Information sharing

when there is a change: Each router sends out information about the neighbors when there ischange.

### ALGORITHM:

1. Createasimulatorobject
2. Definedifferentcolorsfordifferentdataflows
3. Open a nam trace file and define finish procedure then close the trace file, and executenamon tracefile.
4. Createn numberof nodes usingforloop
5. Createduplexlinksbetweenthenodes
6. SetupUDPConnectionbetweenn(0)andn(5)
7. SetupanotherUDPconnectionbetweenn(1)andn(5)
8. ApplyCBR Trafficover both UDP connections
9. ChooseLinkstateroutingprotocoltotransmitdatafromsendertoreceiver.
10. Scheduleevents andruntheprogram.

### Program:

setns[newSimulator]

setnf[openout.namw]

$nsnamtrace-all$nf

settr[openout.trw]

$nstrace-all$tr

procfinish{}{

globalnfnstr

$nsflush-trace

close$tr

execnamout.nam&

exit0

}

set n0 [$ns node]set n1 [$ns node]set n2 [$ns node]setn3[$nsnode]

$nsduplex-link$n0$n110Mb10msDropTail

$nsduplex-link$n1$n310Mb10msDropTail

$nsduplex-link$n2$n110Mb10msDropTail

$nsduplex-link-op$n0$n1orientright-down

$nsduplex-link-op$n1$n3orientright

$nsduplex-link-op$n2$n1 orientright-up

settcp[newAgent/TCP]

$nsattach-agent$n0$tcp

setftp[newApplication/FTP]

$ftpattach-agent$tcp

setsink[newAgent/TCPSink]

$nsattach-agent$n3$sink

setudp[newAgent/UDP]

$nsattach-agent$n2$udp

setcbr[newApplication/Traffic/CBR]

$cbrattach-agent$udp

setnull[newAgent/Null]

$nsattach-agent$n3$null

$nsconnect$tcp$sink

$nsconnect$udp$null

$nsrtmodel-at1.0down$n1$n3

$nsrtmodel-at2.0up$n1$n3

$nsrtprotoLS

$nsat0.0"$ftpstart"

$nsat0.0 "$cbrstart"

$nsat5.0"finish"

$nsrun

**Result:**

ThustheprogramforcreatingSimulationofDistanceVector/LinkState

Routingwasimplemented.

### EXPERIMENT 7

### Implementation of Stop and Wait Protocol and Sliding WindowProtocol.

**STOPANDWAITPROTOCOL**

**//SENDER//**import java.io.\*;importjava.net.\*;

importjava.util.Scanner;classstopwaitsender

{

publicstatic void main(Stringargs[]) throwsException

{

stopwaitsendersws=newstopwaitsender();sws.run();

}

publicvoidrun() throws Exception

{

Scanner sc=new Scanner(System.in);System.out.println(“Enter no of frames to be sent:”);intn=sc.nextInt();

Socketmyskt=newSocket(“localhost”,9999);

PrintStreammyps=newPrintStream(myskt.getOutputStream());for(inti=0;i<=n;)

{

if(i==n)

{

myps.println(“exit”);break;

}

System.out.println(“Frameno“+i+”issent”);myps.println(i);

BufferedReader bf=new BufferedReader(new InputStreamReader(myskt.getInputStream()));Stringack=bf.readLine();

if(ack!=null)

{

System.out.println(“AcknowledgementwasReceivedfromreceiver”);i++;

Thread.sleep(4000);

}

else

{

myps.println(i);

}

}

}

}

//RECEIVER//import java.io.\*;import java.net.\*;classstopwaitreceiver

{

publicstatic voidmain(Stringargs[])throws Exception

{

stopwaitreceiverswr=newstopwaitreceiver();swr.run();

}

publicvoidrun() throws Exception

{

String temp=”any message”,str=”exit”;ServerSocket myss=new ServerSocket(9999);Socketss\_accept=myss.accept();BufferedReaderss\_bf=newBufferedReader(new

InputStreamReader(ss\_accept.getInputStream()));

PrintStreammyps=newPrintStream(ss\_accept.getOutputStream());while(temp.compareTo(str)!=0)

{

Thread.sleep(1000);temp=ss\_bf.readLine();if(temp.compareTo(str)==0)

{break;}

System.out.println(“Frame “+temp+” was received”);Thread.sleep(500);

myps.println(“Received”);

}

System.out.println(“ALLFRAMESWERERECEIVEDSUCCESSFULLY”);

}

}

**OUTPUTFORSENDER:**

C:\javaprog>javacstopwaitsender.javaC:\javaprog>javastopwaitsenderEnterno offrames to besent:

4

Frameno0issent

AcknowledgementwasReceivedfromreceiverFrameno1 is sent

AcknowledgementwasReceivedfromreceiverFrameno2 is sent

AcknowledgementwasReceivedfromreceiverFrameno3 is sent

Acknowledgementwas Receivedfromreceiver

OUTPUT FORRECEIVER:

C:\javaprog>javacstopwaitreceiver.javaC:\javaprog>java stopwaitreceiverFrame0was received

Frame 1 was receivedFrame 2 was receivedFrame3wasreceived

ALLFRAMESWERERECEIVEDSUCCESSFULLY

//import java.lang.System;importjava.net.\*;

import java.io.\*;classbitserver

{

### SLIDINGWINDOWPROTOCOL

import java.lang.System;importjava.net.\*;importjava.io.\*;

import java.math.\*;classbitclient

publicstaticvoidmain(String[]args)

{try

{BufferedInputStream in;ServerSocket Serversocket=newServerSocket(500);

System.out.println("waiting for connection");Socket client=Serversocket.accept();System.out.println("received request for sendingframes");

in=newBufferedInputStream(client.getInputStream()); DataOutputStream out=newDataOutputStream(client.getOutputStream());int p=in.read();System.out.println("sending ");

for(inti=1;i<=p;++i)

{System.out.println("sending frame no"+i);out.write(i);

out.flush();

System.out.println("waiting for acknowledge");Thread.sleep(5000);

inta=in.read();

System.out.println("received acknowledge forframeno:"+i+"as"+a);

}

out.flush();

in.close();

out.close();client.close();Serversocket.close();

System.out.println("quiting");

}catch(IOExceptione)

{System.out.println(e);

}

catch(InterruptedExceptione)

{}

}

}

{

publicstaticvoidmain(Stringa[])

{

try

{InetAddressaddr=InetAddress.getByName("Localhost");System.out.println(addr);

Socket connection=new Socket(addr,500);DataOutputStream out=newDataOutputStream(connection.getOutputStream()

);

BufferedInputStream in=newBufferedInputStream(connection.getInputStream()

);

BufferedInputStream inn=newBufferedInputStream(connection.getInputStream()

);

BufferedReader ki=new BufferedReader(newInputStreamReader(System.in));

int flag=0;System.out.println("connect");

System.out.println("enter the no of frames to berequestedtoserver:");

int c=Integer.parseInt(ki.readLine());out.write(c);

out.flush();int i,jj=0;while(jj<c)

{i=in.read();

System.out.println("received frame no"+i);System.out.println("sending acknowledgement forframeno"+i);

out.write(i);

out.flush();

jj++;

}

out.flush();

in.close();

inn.close();

out.close();

System.out.println("quiting");

}

catch(Exceptione)

{System.out.println(e);

}}}

**OUTPUTBITCLIENT**

Localhost/127.0.0.1connect

enter the no of frames to be requested to server:4receivedframeno1

sending acknowledgement for frame no1receivedframeno2

sending acknowledgement for frame no2receivedframeno3

sending acknowledgement for frame no3receivedframeno4

sendingacknowledgementforframeno4

**BITSERVER**

waitingfor connection

receivedrequestforsendingframessending....

.sending frame no1waitingforacknowledge

received acknowledge for frame no:1as1sending frame no2waiting for acknowledgereceived acknowledge for frame no:2as2sendingframeno3

waitingforacknowledge

received acknowledge for frame no:3as3sendingframeno4

waitingforacknowledge

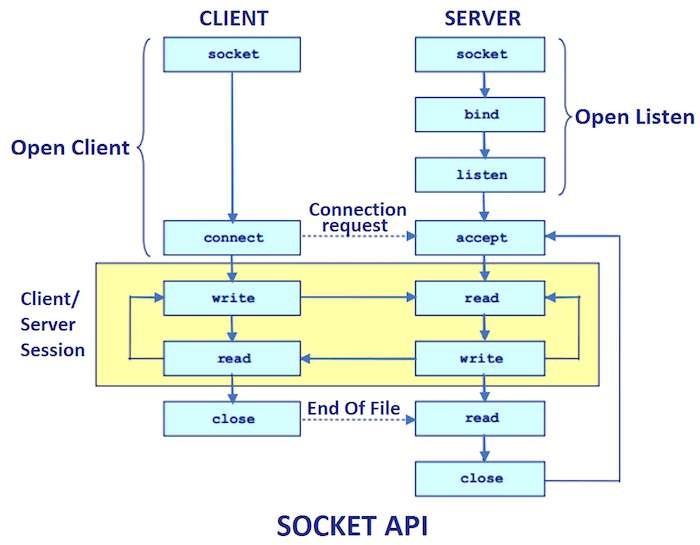
receivedacknowledge forframeno:4as4

### EXPERIMENT 8

### Study of Socket Programming and Client–Server model

Java Socket programming is used for communication between the applications running ondifferent JRE. Java Socket programming can be connection-oriented or connection-less. Theclient in socket programming must know two information: IP Address of Server, and Portnumber.

Here, we are going to make one-way client and server communication. In this application,client sends a message to the server, server reads the message and prints it. Here, two classesare being used: Socket and ServerSocket. The Socket class is used to communicate client andserver. Through this class, we can read and write message. The ServerSocket class is used atserver-side. The accept() method of ServerSocket class blocks the console until the client isconnected. After the successful connection of client, it returns the instance of Socket at server-side.



Socketclass

A socket is simply an endpoint for communications between the machines. The Socket classcanbeusedtocreate asocket.

Importantmethods

|  |  |
| --- | --- |
| **Method Description** | |
| 1)publicInputStreamgetInputStream() | returnstheInputStreamattachedwiththissocket. |
| 2)publicOutputStreamgetOutputStream() | returnstheOutputStream attachedwiththissocket. |
| 3)publicsynchronizedvoidclose() | closesthissocket |

ServerSocketclass

The ServerSocket class can be used to create a server socket. This object is used to establishcommunicationwith theclients.

Importantmethods

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** |  |  | **Description** |
| 1)publicSocketaccept() | | | returnsthesocketandestablishaconnectionbetweenserverandclient. |
| 2) publicclose() | synchronized | void | closestheserversocket. |

ExampleofJavaSocketProgramming

**CreatingServer:**

To create the server application, we need to create the instance of ServerSocketclass. Here,weareusing6666portnumberforthecommunicationbetweentheclientandserver.Youmay also choose any other port number. The accept() method waits for the client. If clientsconnectswith thegiven port number,it returns an instanceof Socket.

1. ServerSocketss=**new**ServerSocket(6666);
2. Sockets=ss.accept();//establishesconnectionandwaitsfortheclient

**CreatingClient:**

To create the client application, we need to create the instance of Socket class. Here, we needto pass the IP address or hostname of the Server and a port number. Here, we are using"localhost"because our server is runningon samesystem.

1. Sockets=**new**Socket("localhost",6666);

Let'sseeasimpleofJavasocketprogrammingwhereclientsendsatextandserverreceivesandprints it.

|  |  |
| --- | --- |
| *File: MyServer.java*import java.io.\*;importjava.net.\*; | *File: MyClient.java*import java.io.\*;importjava.net.\*; |
| publicclassMyServer{  publicstaticvoidmain(String[]args){try{  ServerSocket ss=new ServerSocket(6666);Sockets=ss.accept();//establishesconnection | publicclassMyClient{  publicstaticvoidmain(String[]args){try{  Sockets=newSocket("localhost",6666); |
| DataInputStream dis=newDataInputStream(s.getInputStream()); | DataOutputStream dout=newDataOutputStream(s.getOutputStream()); |
| Stringstr=(String)dis.readUTF();System.out.println("message="+str); | dout.writeUTF("HelloServer");dout.flush(); |
| ss.close(); | dout.close();  s.close(); |
| }catch(Exceptione){System.out.println(e);}  }  } | }catch(Exceptione){System.out.println(e);}  }  } |



### EXPERIMENT 9

### Write a code simulating ARP/RARP protocols.

ThisprogramautomaticallycreatesafilewiththeIPaddressofmachines,theirMACaddressandtype.

ARPprotocolissimulatedbyreadinganIPaddressandreturningtheMACaddress.RARPprotocolis simulated byreadingan MAC addressand returningtheIP address

TheprogramcanbeextendedtoreadanIPAddress/MacAddressandamessageandsendapacketto the specifiedmachine usingTCP/IP orDatagram sockets

import java.io.\*;importjava.util.\*;

publicclassarp\_rarp

{

privatestaticfinalStringCommand= "arp -a";

publicstaticvoidgetARPTable(Stringcmd)throwsException

{

File fp = new File("ARPTable.txt");FileWriterfw=newFileWriter(fp);

BufferedWriter bw = new BufferedWriter(fw);ProcessP=Runtime.getRuntime().exec(cmd);

ScannerS=newScanner(P.getInputStream()).useDelimiter("\\A");

while(S.hasNext())bw.write(S.next());

bw.close();

fw.close();

}

publicstatic void findMAC(Stringip) throwsException

{

File fp = new File("ARPTable.txt");FileReader fr = new FileReader(fp);BufferedReaderbr=newBufferedReader(fr);

Stringline;

while((line=br.readLine())!=null)

{

if(line.contains(ip))

{

System.out.println("InternetAddress PhysicalAddress Type");System.out.println(line);

break;

}

}

if((line==null))

System.out.println("Notfound");

fr.close();

br.close();

}

publicstaticvoidfindIP(Stringmac)throwsException

{

File fp = new File("ARPTable.txt");FileReader fr = new FileReader(fp);BufferedReaderbr=newBufferedReader(fr);

Stringline;

while((line=br.readLine())!=null)

{

if(line.contains(mac))

{

System.out.println("InternetAddress PhysicalAddress Type");System.out.println(line);

break;

}

}

if((line == null))System.out.println("Notfound");

fr.close();

br.close();

}

publicstatic void main(Stringas[]) throws Exception

{

getARPTable(Command);

ScannerS=newScanner(System.in);

System.out.println("ARP Protocol.");System.out.print("Enter IP Address: ");StringIP=S.nextLine();findMAC(IP);

System.out.println("RARP Protocol.");System.out.print("Enter MAC Address: ");StringMAC=S.nextLine();findIP(MAC);

}

}

OUTPUT:

>javacarp\_rarp.java

>java arp\_rarpARPProtocol.

EnterIP Address:10.0.15.253

InternetAddress PhysicalAddress Type10.0.15.253 00-16-76-bd-41-27 dynamic

RARPProtocol.

EnterMACAddress:01-00-5e-00-00-fc

InternetAddress PhysicalAddress Type224.0.0.252 01-00-5e-00-00-fc static

>java arp\_rarpARPProtocol.

Enter IP Address: 10.0.15.121Notfound

RARPProtocol.

Enter MAC Address: 01-00-5e-00-00-ffNotfound

ARPTable.txt

Interface:10.0.15.202 --- 0x3

InternetAddress PhysicalAddress Type

10.0.15.1 54-78-1a-1e-6a-4f dynamic

10.0.15.72 00-13-20-b7-49-c9 dynamic10.0.15.253 00-16-76-bd-41-27 dynamic10.0.15.255 ff-ff-ff-ff-ff-ff static224.0.0.22 01-00-5e-00-00-16 static224.0.0.251 01-00-5e-00-00-fb static224.0.0.252 01-00-5e-00-00-fc static239.255.255.250 01-00-5e-7f-ff-fa static

### EXPERIMENT 10

### (a)Write a code simulating PING commands

**Algorithm**

Step1:starttheprogram.

Step2:Includenecessarypackageinjava.

Step 3: To create a process object p to implement the ping command.Step4: declareoneBufferedReader stream class object.

Step 5: Get thedetails of the server5.1:lengthofthe IPaddress.

* 1. :timerequiredto getthedetails.
  2. : send packets , receive packets and lost packets.5.4:minimum ,maximumand averagetimes.

Step6:printtheresults.

Step7:Stoptheprogram.

**Program:**

import java.io.\*;import java.net.\*;classpingserver

{

publicstaticvoid main(Stringargs[])

{

try

{

Stringstr;

System.out.print(" Enter the IP Address to be Ping : ");BufferedReader buf1=new BufferedReader(newInputStreamReader(System.in));

Stringip=buf1.readLine();Runtime H=Runtime.getRuntime();Processp=H.exec("ping"+ip);

InputStream in=p.getInputStream();BufferedReader buf2=new BufferedReader(newInputStreamReader(in));while((str=buf2.readLine())!=null)

{

System.out.println(""+ str);

}

}

catch(Exceptione)

{

System.out.println(e.getMessage());

}

}

}

**Output:**

Enter the IP address to the ping:192.168.0.1Pinging192.168.0.1:withbytesofdata=32

Reply from 192.168.0.11:bytes=32 time<1ms TTL =128Reply from 192.168.0.11:bytes=32 time<1ms TTL =128Reply from 192.168.0.11:bytes=32 time<1ms TTL =128Replyfrom192.168.0.11:bytes=32time<1msTTL=128

Pingstatistics for192.168.0.1

Packets:sent=4,received=4,lost=0(0%loss),approximateroundtriptimeinmilliseconds:

Minimum=1ms,maximum=4ms,average=2ms

### EXPERIMENT10(b)

### Write a code simulating Trace route commands

import java.io.BufferedReader;importjava.io.InputStreamReader;

publicclasstraceroutecmd

{

publicstaticvoidrunSystemCommand(String command)

{

try

{

Process p = Runtime.getRuntime().exec(command);BufferedReaderinputStream=newBufferedReader(

newInputStreamReader(p.getInputStream()));

Strings = "";

while ((s = inputStream.readLine()) != null)System.out.println(s);

}

catch(Exceptione)

{

}

}

publicstaticvoidmain(String[]args)

{

//Stringip="[www.google.co.in"](http://www.google.co.in/);

//Stringip ="127.0.0.1";

String ip = "[www.drranurekha.com](http://www.drranurekha.com/)";runSystemCommand("tracert"+ip);

}

}

OUTPUT:

>javactraceroutecmd.java

>javatraceroutecmd

Tracingroute todrranurekha.com [160.153.137.167] over amaximum of30 hops:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 <1ms | | <1ms | | <1ms 10.0.15.1 | |
| 2 <1ms | | <1ms | | <1ms 10.0.0.15 | |
| 3 1ms | | 1ms | | 1ms210.212.247.209 | |
| 4 2ms | | 1ms | | 1ms172.24.75.102 | |
| 5 | \* | \* | 21 ms218.248.235.217 | | |
| 6 | \* | \* | 12 ms218.248.235.218 | | |
| 7 21ms | | 21ms | | 21ms121.244.37.253.static.chennai.vsnl.net.in[121.244.37.253] | |
| 8 \* | | \* \* | | Requesttimedout. | |
| 9 49ms | | 49ms | | 49ms172.25.81.134 | |
| 10 50ms | | 50ms | | 70msix-ae-0-4.tcore1.mlv-mumbai.as6453.net[180.87.38.5] | |
| 11 | 165ms 165 ms | | | 165ms | if-ae-9-5.tcore1.wyn-marseille.as6453.net[80.231.217.17] |
| 12 | 172ms 171 ms | | | 171ms | if-ae-8-1600.tcore1.pye-paris.as6453.net[80.231.217.6] |
| 13 | 171ms 171 ms | | | 171ms | if-ae-15-2.tcore1.av2-amsterdam.as6453.net |

[195.219.194.145]

14 175ms175ms175 ms195.219.194.2

15171ms170 ms170 mspo72.bbsa0201-01.bbn.mgmt.ams1.gdg [188.121.33.74]

16 170ms169ms169 ms10.241.131.203

17 175ms175ms175 ms10.253.1.1

18166 ms166 ms166 ms10.253.130.9

19173 ms173 ms173 ms10.253.130.3

20169 ms169 ms169 ms10.253.130.5

21169 ms169 ms169 msip-160-153-137-167.ip.secureserver.net [160.153.137.167]Tracecomplete.

### EXPERIMENT11

### Create a socket for HTTP for web page upload and download.

**Algorithm**

* 1. Starttheprogram.
  2. Gettheframe sizefrom theuser
  3. To create the frame based on the user request.4.Tosendframestoserverfromtheclientside.

1. If your frames reach the server it will send ACK signal to client otherwise it will sendNACKsignal to client.
2. Stop the programProgram:

### Client

import javax.swing.\*;importjava.net.\*;import java.awt.image.\*;import javax.imageio.\*;importjava.io.\*;

import java.awt.image.BufferedImage; import java.io.ByteArrayOutputStream; importjava.io.File;

import java.io.IOException; import javax.imageio.ImageIO;publicclass Client{

publicstatic voidmain(Stringargs[])throws Exception{Socket soc;

BufferedImage img = null;soc=newSocket("localhost",4000);

System.out.println("Client is running. ");try{

System.out.println("Readingimagefromdisk.");

img = ImageIO.read(new File("digital\_image\_processing.jpg")); ByteArrayOutputStreambaos=newByteArrayOutputStream();

ImageIO.write(img, "jpg", baos);baos.flush();

byte[]bytes=baos.toByteArray();baos.close();

System.out.println("Sending image to server. ");OutputStream out = soc.getOutputStream();DataOutputStream dos = new DataOutputStream(out);dos.writeInt(bytes.length);

dos.write(bytes, 0, bytes.length);System.out.println("Imagesenttoserver.");

dos.close();

out.close();

}catch (Exception e) { System.out.println("Exception: " + e.getMessage());soc.close();

}

soc.close();

}

}

### Server

import java.net.\*;importjava.io.\*;

import java.awt.image.\*;import javax.imageio.\*;importjavax.swing.\*;

classServer{

public static void main(String args[]) throws Exception{ServerSocketserver=null;

Socketsocket;

server=new ServerSocket(4000);System.out.println("ServerWaitingforimage");

socket=server.accept(); System.out.println("Client connected.");InputStreamin= socket.getInputStream();

DataInputStream dis = new DataInputStream(in);intlen =dis.readInt();

System.out.println("Image Size: " + len/1024 + "KB"); byte[] data = new byte[len];dis.readFully(data);

dis.close();

in.close();

InputStream ian = new ByteArrayInputStream(data);BufferedImage bImage =ImageIO.read(ian);JFramef=new JFrame("Server");

ImageIcon icon = new ImageIcon(bImage);JLabell =new JLabel();

l.setIcon(icon);f.add(l);

f.pack();

f.setVisible(true); } }

### EXPERIMENT12

### Write a program to implement RPC(Remote Procedure Call)

A remote procedure call is an inter-process communication technique that is used for client-serverbased applications.It isalso known as a subroutinecall orafunctioncall.

A client has a request message that the RPC translates and sends to the server. This requestmay be a procedure or a function call to a remote server. When the server receives the request,it sends the required response back to the client. The client is blocked while the server isprocessingthe call and onlyresumed execution afterthe server is finished.

Thesequenceof events inaremoteprocedurecallare given asfollows−

* Theclientstub iscalledbythe client.
* Theclientstubmakesasystemcalltosendthemessagetotheserverandputstheparametersin the message.
* Themessageissent fromtheclientto theserverbythe client’soperatingsystem.
* Themessageis passed totheserver stub bythe server operatingsystem.
* Theparametersare removedfromthemessagebythe serverstub.
* Then,the server procedureis called bythe serverstub.

# HelloWorld.java

packagerpc\_helloworld;

importjavax.jws.WebMethod;importjavax.jws.WebService;importjavax.jws.soap.SOAPBinding;

import javax.jws.soap.SOAPBinding.Style;@WebService

@SOAPBinding(style = Style.RPC)public interface HelloWorld {

@WebMethod String getHelloWorld(String name);

}

# HelloWorldImpl.java

packagerpc\_helloworld;

import javax.jws.WebService;@WebService(endpointInterface ="rpc\_helloworld.HelloWorld")

public class HelloWorldImpl implements HelloWorld{@Override

public String getHelloWorld(String name) {return name;

}

}

# Publisher.java

packagerpc\_helloworld;

import javax.xml.ws.Endpoint;public class Publisher {

public static void main(String[] args) {

Endpoint.publish("http://localhost:7779/ws/hello",new HelloWorldImpl());

}

}

# rpc\_helloworld.java

packagerpc\_helloworld;

import java.net.MalformedURLException;import java.net.URL;

import java.util.logging.Level;import java.util.logging.Logger;import javax.xml.namespace.QName;import javax.xml.ws.Service;public class RPC\_HelloWorld {

public static void main(String[] args) {try {

//Refer to wsdl documentURL url = new

URL("http://localhost:7779/ws/hello?wsdl");

//Refer to wsdl documentQName qname = new

QName("http://rpc\_helloworld/","HelloWorldImplService");

Service service = Service.create(url,

qname);

HelloWorld hello =

service.getPort(HelloWorld.class);

System.out.println(hello.getHelloWorld("HelloWorld!"));

} catch (MalformedURLException ex) {

System.out.println("WSDL document urlerror: " + ex);

}

}

}

Aftercreateallthesefiles,youneedtorunyourPublisher.Javaandthengotoyourbrowserandtypethefollowing:

http://localhost:7779/ws/hello?wsdl

Then you will get the response in XML format. After that you need to copy the text that assignfortargetNamespace.

ThenpastethetextinyourClientsidefile,astheQNamefirstparameter.Nowrunyourprogramandyou willgettheoutput.

[https://sivakumar-prasanth.medium.com/java-rpc-remote-procedure-call-99cfaca34c36]

### EXPERIMENT13

### Implementation of Sub netting.

import java.util.Scanner;classSubnet{

public static void main(String args[]){Scanner sc = new Scanner(System.in);System.out.print(“Enter the ip address: “);Stringip = sc.nextLine();

String split\_ip[] = ip.split(“\\.”); //SPlit the string after every .Stringsplit\_bip[]=new String[4]; //split binaryip

String bip = “”;for(inti=0;i<4;i++){

split\_bip[i]=appendZeros(Integer.toBinaryString(Integer.parseInt(split\_ip[i])));//“18”=>18

=>10010 =>00010010

bip+=split\_bip[i];

}

System.out.println(“IP in binary is “+bip);System.out.print(“Enter the number of addresses: “);intn =sc.nextInt();

//Calculationofmask

int bits = (int)Math.ceil(Math.log(n)/Math.log(2)); /\*eg if address = 120, log 120/log 2 giveslogto thebase2=>6.9068, ceil gives us upperinteger \*/

System.out.println(“Number of bits required for address = “+bits);intmask =32-bits;

System.out.println(“Thesubnetmaskis=“+mask);

//Calculation of first address and last addressintfbip[]=new int[32];

for(int i=0; i<32;i++) fbip[i] = (int)bip.charAt(i)-48; //convert cahracter 0,1 to integer 0,1for(inti=31;i>31-bits;i–)//Get first address byANDinglast n bits with 0

fbip[i] &=0;

String fip[] = {“”,””,””,””};for(inti=0;i<32;i++)

fip[i/8] = new String(fip[i/8]+fbip[i]);System.out.print(“First address is = “);for(int i=0;i<4;i++){System.out.print(Integer.parseInt(fip[i],2));if(i!=3)System.out.print(“.”);

}

System.out.println();

intlbip[] =newint[32];

for(int i=0; i<32;i++) lbip[i] = (int)bip.charAt(i)-48; //convert cahracter 0,1 to integer 0,1for(inti=31;i>31-bits;i–)//Get last address byORinglast n bits with 1

lbip[i]|=1;

String lip[] = {“”,””,””,””};for(inti=0;i<32;i++)

lip[i/8] = new String(lip[i/8]+lbip[i]);System.out.print(“Last address is = “);for(int i=0;i<4;i++){System.out.print(Integer.parseInt(lip[i],2));if(i!=3)System.out.print(“.”);

}

System.out.println();

}

static String appendZeros(String s){String temp = new String(“00000000”);returntemp.substring(s.length())+s;

}

}

/\***Output**

Entertheipaddress: 100.110.150.10

IPin binaryis 01100100011011101001011000001010

Enter the number of addresses: 7Number of bits required for address = 3Thesubnet mask is =29

First address is = 100.110.150.8Lastaddressis=100.110.150.15

\*/

### EXPERIMENT 13a

### IMPLEMENTATION OF TCP/IP ECHO.ALGORITHM

**Server**

1. Createaserver socketandbind ittoport.
2. Listenfornewconnectionandwhenaconnectionarrives,acceptit.
3. Readthedata fromclient.
4. Echothedata backto theclient.
5. Repeatsteps4-5until„bye‟or„null‟is read.
6. Closeallstreams.
7. Closetheserversocket.
8. Stop.

### Client

1. Createaclientsocketandconnectit totheserver‟sportnumber.
2. Getinputfromuser.
3. Ifequaltobyeornull,thengoto step7.
4. Senduserdatatotheserver.
5. Displaythedataechoedbytheserver.
6. Repeatsteps2-4.
7. Closetheinputand outputstreams.
8. Closetheclient socket.
9. Stop.

### OUTPUT

Server:

$javactcpechoserver.java

$javatcpechoserver

ServerReadyClientConnected Client [ hello ]

Client [ how are you ] Client [ i am fine ] Client [ ok ]ClientDisconnected

Client:

$javactcpechoclient.java

$ java tcpechoclientType"bye"toquit

Enter msg to server : helloServer[ hello ]

Enter msg to server : how are youServer[ howareyou ]

Enter msg to server : i am fineServer[ i am fine ]

Enter msg to server : okServer[ ok ]

Entermsgtoserver :bye

/ TCP Echo Server--tcpechoserver.javaimportjava.net.\*;

importjava.io.\*;

publicclasstcpechoserver

{

publicstaticvoidmain(String[]arg)throwsIOException

{

ServerSocket sock = null;BufferedReader fromClient = null;OutputStreamWritertoClient=null;Socketclient =null;

try

{

sock = new ServerSocket(4000);System.out.println("Server Ready");client = sock.accept();System.out.println("ClientConnected");fromClient=newBufferedReader(new

InputStreamReader(client.getInputStream()));toClient = newOutputStreamWriter(client.getOutputStream());Stringline;

while(true)

{

line=fromClient.readLine();

if((line==null)||line.equals("bye"))break;

System.out.println("Client["+line+"]");toClient.write("Server [ "+ line +" ]\n");toClient.flush();

}

fromClient.close();toClient.close();client.close();sock.close();

System.out.println("ClientDisconnected");

}

catch(IOExceptionioe)

{

System.err.println(ioe);

}

}

}

//TCP Echo Client--tcpechoclient.javaimportjava.net.\*;

importjava.io.\*;

publicclasstcpechoclient

{

public static void main(String[] args) throwsIOException

{

BufferedReader fromServer = null, fromUser =null;

PrintWriter toServer = null;Socketsock =null;

try

{

if (args.length == 0)sock=new

Socket(InetAddress.getLocalHost(),4000);else

sock = newSocket(InetAddress.getByName(args[0]),4000);fromServer = new BufferedReader(newInputStreamReader(sock.getInputStream()));fromUser = new BufferedReader(newInputStreamReader(System.in));

toServer = newPrintWriter(sock.getOutputStream(),true);String Usrmsg, Srvmsg;System.out.println("Type\"bye\"toquit");while(true)

{

System.out.print("Enter msg to server : ");Usrmsg=fromUser.readLine();

if(Usrmsg==null ||Usrmsg.equals("bye"))

{

toServer.println("bye");break;

}

elsetoServer.println(Usrmsg);

Srvmsg=fromServer.readLine();System.out.println(Srvmsg);

}

fromUser.close();fromServer.close();toServer.close();sock.close();

}

catch(IOExceptionioe)

{

System.err.println(ioe);

}

|  |  |  |  |
| --- | --- | --- | --- |
| **EXPERIMENT13b**  **IMPLEMENTATION OF TCP/IP Chat** | | | |
|  | //AJavaprogramforaClientimportjava.net.\*;  importjava.io.\*;  publicclassClient  {  // initialize socket and input outputstreams  privateSocketsocket = null;private DataInputStreaminput= null;privateDataOutputStreamout =null;  // constructor to put ip address and portpublicClient(Stringaddress, intport)  {  // establish a connectiontry  {  socket = new Socket(address, port);System.out.println("Connected");  //takesinputfromterminalinput=new  DataInputStream(System.in);  //sendsoutputtothesocketout =new  DataOutputStream(socket.getOutputStream()  );  }  catch(UnknownHostExceptionu)  {  System.out.println(u);  }  catch(IOExceptioni)  {  System.out.println(i);  }  //stringtoreadmessagefrominputStringline="";  //keepreadinguntil "Over"isinputwhile(!line.equals("Over"))  {  try  {  line=input.readLine();out.writeUTF(line);  }  catch(IOExceptioni)  { | //AJavaprogramforaServerimportjava.net.\*;  importjava.io.\*;  publicclassServer  {  //initialize socket and input streamprivateSocket socket= null;privateServerSocket server= null;privateDataInputStreamin =null;  //constructorwithportpublicServer(intport)  {  //startsserverandwaitsforaconnectiontry  {  server = new ServerSocket(port);System.out.println("Serverstarted");  System.out.println("Waitingforaclient...");  socket = server.accept();System.out.println("Clientaccepted");  //takesinputfromtheclientsocketin= new DataInputStream(  newBufferedInputStream(socket.getInputStream())  );  Stringline= "";  // reads message from client until"Over"is sent  while(!line.equals("Over"))  {  try  {  line = in.readUTF();System.out.println(line);  }  catch(IOExceptioni)  {  System.out.println(i);  }  }  System.out.println("Closingconnection"); |  |
|  | System.out.println(i); | //closeconnection |  |

|  |  |
| --- | --- |
| } | socket.close(); |
| } | in.close(); |
|  | } |
| //close theconnection | catch(IOExceptioni) |
| try | { |
| { | System.out.println(i); |
| input.close(); | } |
| out.close(); | } |
| socket.close(); |  |
| } | publicstatic voidmain(Stringargs[]) |
| catch(IOExceptioni) | { |
| { | Serverserver=new Server(5000); |
| System.out.println(i); | } |
| } | } |
| } |  |
| publicstatic voidmain(Stringargs[]) |  |
| { |  |
| Clientclient=newClient("127.0.0.1", |  |
| 5000); |  |
| } |  |
| } |  |
|  | $javaServer |
| $javaClient | Hello |
| Hello | Imademyfirstsocketconnection |
| Imademyfirstsocket | Over |
| connection | Closingconnection |
| Over |  |

### EXPERIMENT13c

### IMPLEMENTATIONOFTCP/IPSOCKET FILETRANSFER

**AlgorithmServer**

Step1: Import java packages and create class file server.Step2: Create a new server socket and bind it to the port.Step3:Accept theclientconnection

Step4: Get the file name and stored into the BufferedReader.Step5:Create anewobject class fileandrealine.

Step6: If file is exists then FileReader read the content until EOF is reached.Step7:Stop theprogram.

Client

Step1: Import java packages and create class file server.Step2: Create a new server socket and bind it to the port.Step3:Nowconnection is established.

Step4: The object of a BufferReader class is used for storing data content which has beenretrievedfrom socket object.

Step5 The connection is closed.Step6:Stop theprogram.

### ProgramFileServer:

import java.io.BufferedInputStream;importjava.io.File;

importjava.io.FileInputStream;import java.io.OutputStream;import java.net.InetAddress;import java.net.ServerSocket;importjava.net.Socket;

publicclassFileServer

{

publicstaticvoidmain(String[] args)throwsException

{

//InitializeSockets

ServerSocket ssock = new ServerSocket(5000);Socketsocket = ssock.accept();

//TheInetAddressspecification

InetAddress IA=InetAddress.getByName("localhost");

//Specifythefile

File file = new File("e:\\Bookmarks.html");FileInputStreamfis=newFileInputStream(file);

BufferedInputStreambis=newBufferedInputStream(fis);

//Getsocket'soutput stream

OutputStreamos=socket.getOutputStream();

//Read File Contents into contents arraybyte[] contents;

long fileLength = file.length();longcurrent=0;

long start = System.nanoTime();while(current!=fileLength){

intsize =10000;

if(fileLength - current >= size)current+=size;

else{

size = (int)(fileLength - current);current=fileLength;

}

contents = new byte[size];bis.read(contents, 0, size);os.write(contents);

System.out.print("Sendingfile..."+(current\*100)/fileLength+"%complete!");

}

os.flush();

//File transfer done. Close the socket connection!socket.close();

ssock.close();

System.out.println("Filesentsuccesfully!");

} }

### FileClient

import java.io.BufferedOutputStream;import java.io.FileOutputStream;importjava.io.InputStream;

import java.net.InetAddress;importjava.net.Socket;

publicclassFileClient {

publicstaticvoidmain(String[] args)throwsException{

//Initializesocket

Socket socket = new Socket(InetAddress.getByName("localhost"), 5000);byte[] contents = new byte[10000];

//InitializetheFileOutputStream totheoutputfile'sfull path.

FileOutputStream fos = new FileOutputStream("e:\\Bookmarks1.html");BufferedOutputStream bos = new BufferedOutputStream(fos);InputStream is =socket.getInputStream();

//No of bytes read in one read() callint bytesRead=0;

while((bytesRead=is.read(contents))!=-1)bos.write(contents,0,bytesRead);

bos.flush();socket.close();

System.out.println("Filesavedsuccessfully!");

}

}

**Output**

**server**

E:\nwlab>java FileServerSending file ... 9% complete!Sending file ... 19% complete!Sending file ... 28% complete!Sending file ... 38% complete!Sending file ... 47% complete!Sending file ... 57% complete!Sending file ... 66% complete!Sending file ... 76% complete!Sending file ... 86% complete!Sendingfile...95%complete!

Sending file ... 100% complete!Filesent successfully!

E:\nwlab>clientE:\nwlab>javaFileClientFilesavedsuccessfully!

E:\nwlab>

### EXPERIMENT14

### Program to implement DNS in java

import java.net.\*;import java.io.\*;importjava.util.\*;

publicclassDNS

{

publicstaticvoidmain(String[]args)

{

int n;

BufferedReaderin=newBufferedReader(newInputStreamReader(System.in));do

{

System.out.println("\n Menu:\n1.DNS2.ReverseDNS3.Exit \n");System.out.println("\nEnteryour choice");

n=Integer.parseInt(System.console().readLine());if(n==1)

{

try

{

System.out.println("\nEnterHostName");Stringhname=in.readLine();

InetAddressaddress;

address=InetAddress.getByName(hname);System.out.println("HostName:"+address.getHostName());System.out.println("IP:"+address.getHostAddress());

}

catch(IOExceptionioe)

{

ioe.printStackTrace();

}

}

if(n==2)

{

try

{

System.out.println("\nEnterIPaddress");Stringipstr =in.readLine();

InetAddress ia = InetAddress.getByName(ipstr);System.out.println("IP: "+ipstr);System.out.println("HostName:"+ia.getHostName());

}

catch(IOExceptionioe)

{

ioe.printStackTrace();

}

}

}while(!(n==3));

}

}

